



Universidade Nova de Lisboa
OMNIS CIVITAS CONTRA SE DIVISA NON STABIT
Faculdade de Ciências e Tecnologia



Laboratório Remoto

para Monitorização e Controlo baseado

em *MATLAB/Simulink* e *LabVIEW*

Luís Filipe Fernandes Brazão da Silva

Dissertação submetida à Universidade Nova de Lisboa para obtenção do grau de
Mestre em Engenharia Electrotécnica e Computadores,
sob a orientação do Professor Doutor Luís Brito Palma

Portugal, Lisboa 2011

AGRADECIMENTOS

Ao Professor Luís Palma pela orientação e sobretudo pela paciência, dedicação e profissionalismo que demonstrou no acompanhamento do trabalho.

Aos meus pais, Ângela e Amílcar, ao meu irmão, Pedro, aos meus avós Gilda, Cesária e Sílvia, aos meus tios Carlos e Gália e ao meu primo Nuno, que sempre me incentivaram na realização de meus estudos e cujo apoio foi essencial para concluir a minha formação académica.

Aos amigos da Madeira Rodolfo Gouveia, Orlando Fernandes, Rui Ferreira, Miguel Fernandes, Diogo Mota, Catarina Ladeira, Maria João, Filipa Morgado, Sérgio Balona e Carolina Fernandes, entre outros, pela vossa amizade e por estarem presentes sempre que precisei.

Aos amigos Nuno Matos, Tiago Raposo, André Garrido, Miguel Luís, Carlos Dias, Pedro Domingos, Pedro Dantas e Pedro Lopes pelo apoio e ajuda prestada, quer durante o curso, quer noutras situações pessoais, em que se revelaram bons amigos.

À Catarina pela paciência, companheirismo, dedicação, apoio e amizade que teve durante esta jornada.

À família da Catarina, Cristina Gonçalves e Bernardo Gonçalves pelo apoio e carinho.

E por fim, a todos os que de alguma forma contribuíram para a realização e para a finalização deste trabalho.

ÍNDICE

| | |
|---|------|
| LISTA DE FIGURAS..... | V |
| LISTA DE TABELAS..... | VII |
| RESUMO | IX |
| ABSTRACT | XI |
| LISTA DE ACRÓNIMOS | XIII |
| 1 INTRODUÇÃO | 1 |
| 1.1 Motivações | 1 |
| 1.2 Objectivos e Contribuições | 2 |
| 1.3 Estrutura da Dissertação | 3 |
| 2 LABORATÓRIOS REMOTOS..... | 5 |
| 2.1 Definição de laboratórios remotos | 5 |
| 2.2 Desafios face aos laboratórios tradicionais..... | 7 |
| 2.3 Vantagens e desvantagens..... | 8 |
| 2.4 Exemplos de software de aquisição e processamento de dados..... | 9 |
| 2.5 Diferentes abordagens para laboratórios remotos..... | 10 |
| 3 ARQUITECTURA PROPOSTA E CONCEITOS TEÓRICOS..... | 11 |
| 3.1 Arquitectura proposta..... | 11 |
| 3.2 Simulation Interface Toolkit (SIT) | 14 |
| 3.3 Funções “S” para integração <i>MATLAB/Simulink</i> | 20 |
| 3.3.1 Funções “S” (<i>S-Functions</i>) | 20 |
| 3.3.2 Implementação de Funções “S” | 24 |
| 3.4 Transformada de Hilbert | 27 |
| 4 IMPLEMENTAÇÃO PRÁTICA E RESULTADOS EXPERIMENTAIS | 29 |
| 4.1 Processo dinâmico (“ <i>three-tank benchmark</i> ”)..... | 29 |
| 4.1.1 Modelo dinâmico do sistema | 30 |
| 4.1.2 O problema do <i>benchmark</i> | 33 |
| 4.2 Simulador para modelo de três tanques (“ <i>three-tank benchmark</i> ”) | 33 |
| 4.3 RESULTADOS EXPERIMENTAIS | 43 |
| 4.3.1 Descrição da Simulação..... | 43 |
| 4.3.2 Resultados Obtidos | 44 |
| 5 CONCLUSÕES..... | 49 |

| | | |
|---|----------------------------|----|
| 5.1 | Considerações Finais | 49 |
| 5.2 | Trabalho Futuro..... | 50 |
| Apêndice A – Simulador de processo PCT-9..... | | 53 |
| Apêndice B – Simulador de processo de três tanques da Amira | | 61 |
| REFERÊNCIAS BIBLIOGRÁFICAS | | 65 |

LISTA DE FIGURAS

| | |
|---|----|
| Figura 2.1 - Estrutura de um possível laboratório remoto [Marchezan, 2004]. | 6 |
| Figura 2.2 - Laboratório tradicional. | 8 |
| Figura 3.1 - Arquitectura conceptual proposta. | 12 |
| Figura 3.2 - Arquitectura proposta para laboratório remoto com integração do <i>MATLAB/Simulink</i> e do <i>LabVIEW</i> . | 13 |
| Figura 3.3 - Ligação de várias <i>Vis</i> e de vários modelos <i>Simulink</i> . [<i>Mathworks</i>]. | 15 |
| Figura 3.4 - Modelo em <i>Simulink</i> . | 17 |
| Figura 3.5 - Painel frontal do <i>LabVIEW</i> . | 18 |
| Figura 3.6 - SIT Connection Manager. | 19 |
| Figura 3.7 - Bloco Função "S". | 20 |
| Figura 3.8 - Parâmetros do bloco Função "S". | 21 |
| Figura 3.9 - Representação de um bloco do <i>Simulink</i> . | 21 |
| Figura 3.10 - Etapas de uma simulação e correspondência com o valor de <i>flag</i> . | 23 |
| Figura 3.11- Diagrama para a Função "S" <i>gain01</i> . | 24 |
| Figura 3.12 - Ganho variável. | 25 |
| Figura 3.13 - Diferença entre a entrada actual e a anterior. | 26 |
| Figura 4.1 - Sistema de três tanques [Heiming, 2006]. | 30 |
| Figura 4.2 - Modelo de três tanques [Heiming, 2006]. | 33 |
| Figura 4.3 - Simulador em <i>Simulink</i> . | 34 |
| Figura 4.4 - Bloco <i>Level1 – Pump 1 controller</i> . | 35 |
| Figura 4.5 - Bloco <i>Level2 – Pump 2 controller</i> . | 36 |
| Figura 4.6 - Bloco 3Tank System. | 37 |
| Figura 4.7 - Blocos para detecção de oscilações do tanque 1. | 37 |
| Figura 4.8 - Blocos para o cálculo da frequência instantânea. | 39 |
| Figura 4.9 - Painel Frontal do simulador em <i>LabVIEW</i> . | 41 |
| Figura 4.10 - Diagrama de Blocos do simulador em <i>LabVIEW</i> . | 42 |
| Figura 4.11 - Gráficos em <i>MATLAB/Simulink</i> dos resultados da simulação. | 44 |
| Figura 4.12 - Gráficos em <i>LabVIEW</i> . | 45 |
| Figura 4.13 - Gráfico da bomba 1 em: a) <i>Simulink</i> ; b) <i>LabVIEW</i> . | 45 |
| Figura 4.14 - Gráfico do nível do tanque 1 em: a) <i>Simulink</i> ; b) <i>LabVIEW</i> . | 46 |
| Figura 4.15 - Gráfico do nível do tanque 2 em: a) <i>Simulink</i> ; b) <i>LabVIEW</i> . | 46 |
| Figura 4.16 - Gráfico do nível de água no tanque 3 em: a) <i>Simulink</i> ; b) <i>LabVIEW</i> . | 47 |
| Figura 4.17 - Gráfico da detecção de passagens pelo "SetPoint" em: a) <i>Simulink</i> ; b) <i>LabVIEW</i> . | 47 |
| Figura 4.18 - Gráfico da frequência instantânea em: a) <i>Simulink</i> (a azul a frequência instantânea e a verde o erro); b) <i>LabVIEW</i> . | 48 |
| Figura 5.1 - Arquitectura proposta para trabalho futuro. | 51 |
| Figura A.1 - Esquema do processo PCT-9. | 53 |
| Figura A.2 - Processo PCT-9. | 54 |
| Figura A.3 - Electrónica do sensor de nível do tipo potenciómetro. | 54 |
| Figura A.4 - Sensor de nível com bóia. | 55 |

| | |
|--|----|
| Figura A.5 - Sensor (Farnell 178-523) de fluxo de água. | 56 |
| Figura A.6 - Electrónica do conversor F/V..... | 56 |
| Figura A.7 - Painel frontal do simulador do PCT-9 modificado. | 57 |
| Figura A.8 - Diagrama de blocos do simulador do PCT-9 modificado..... | 58 |
| Figura B.1 - Novo processo DTS-200 Amira..... | 61 |
| Figura B.2 - Novo painel frontal para o novo processo de tanques..... | 62 |
| Figura B.3 - Novo diagrama de blocos para o novo processo de tanques. | 63 |

LISTA DE TABELAS

| | |
|---|----|
| Tabela 2.1 - Diferentes abordagens para laboratórios remotos..... | 10 |
| Tabela 3.1 - Parâmetros de entrada de uma Função “S” | 22 |
| Tabela 3.2 - Parâmetros de saída de uma Função “S” | 22 |
| Tabela A.1 - Valores dos testes efectuados. | 55 |

RESUMO

O uso de laboratórios remotos no ensino tem vindo a aumentar na última década. Em Portugal, verificou-se um esforço para introduzir laboratórios remotos no ensino básico, secundário e universitário. Hoje em dia são vastas as oportunidades que a *Internet* oferece no ensino, onde por exemplo são cada vez mais utilizadas técnicas de *e-learning* que permitem aos alunos uma aprendizagem fora do estabelecimento de ensino.

O trabalho de investigação que serviu de base a esta dissertação teve como principal objectivo o desenvolvimento de um sistema de supervisão e controlo de sistemas dinâmicos e a sua aplicação em processos laboratoriais ou industriais. Estes processos apresentam características não lineares e são representativos de sistemas de maior dimensão e complexidade, por exemplo, em processos industriais, sistemas de protecção ambiental ou equipamentos médicos.

Esta dissertação oferece uma arquitectura de um laboratório remoto, onde o termo remoto é utilizado com o objectivo de facilitar aos alunos e professores, através da *Internet*, o acesso a todas as funcionalidades e recursos de um laboratório real. Assim pretende-se eliminar algumas barreiras temporais e geográficas que um laboratório convencional impõe.

Para tal foi desenvolvido um simulador em *MATLAB/Simulink* e em *LabVIEW* do *Benchmark* de um sistema de três tanques. Este simulador permitirá o acesso a utilizadores de diversos lugares, que poderão efectuar a prática de um laboratório remotamente e adquirir uma visão completa do mesmo. Neste simulador foram implementados controladores e detectores de oscilação de forma a proporcionar ao utilizador uma série de ferramentas úteis para a realização do trabalho de laboratório.

ABSTRACT

The use of remote laboratories in education has increased over the last decade. In Portugal, there was an effort to introduce remote labs for basic, secondary and university levels. Nowadays, the *Internet* offers to both students and teachers the opportunities to experience new ways of teaching and learning. As an example, e-learning techniques are becoming a more used tool to allow students to learn outside of the educational facilities.

The research that formed the basis of this thesis is aimed mainly at developing a system of supervision and control of dynamical systems and their application in industrial or laboratory processes. These processes have nonlinear characteristics and are representative of systems of greater size and complexity, for example, in industrial processes, environmental protection systems or equipment.

This thesis proposed an architecture of a remote laboratory, where the term remote is used with the objective of enabling students and teachers, via the *Internet*, to access all functionality and features of a real lab eliminating some time and space barriers that a conventional laboratory imposes.

The objective of this dissertation is to obtain a remote laboratory for testing drivers. For this purpose a simulator was developed in *MATLAB / Simulink* and *LabVIEW* of the Benchmark of a three tanks system. This simulator will allow users to access from different places, to make the practice of a laboratory remotely and gain a comprehensive view of this same laboratory. In this simulator were implemented controllers and oscillation detectors in order to provide the user with a series of useful tools to test drivers.

LISTA DE ACRÓNIMOS

A – Área de secção do tanque cilíndrico

az – Termo de correlação de fluxo

F/V – Frequência / Tensão

g – Constante de gravidade

h – Altura de água no tanque

h_i, h_j – Níveis de água de dois tanques (i e j)

IAE – Integral do erro absoluto (*Integral of Absolute Error*)

K_d – Ganho da parcela derivativa do controlador

K_i – Ganho da parcela integral do controlador

K_p – Ganho da parcela proporcional do controlador

PID – Proporcional Integral Derivativo

PCT – *Process Control Technology*

Q_{in} – Quantidade de água que entra no tanque

Q_{out} – Quantidade de água que sai do tanque

S – Área de secção da válvula de conexão

SIT – *Simulation Interface Toolkit*

SP – *SetPoint*

TCP/IP – *Transmission Control Protocol/ Internet Protocol*

TH – Transformada de Hilbert

V – Volume do tanque

VI – Virtual Instruments

1 INTRODUÇÃO

1.1 Motivações

O ensino à distância sofreu, ao longo dos anos, de várias limitações ao nível da interacção entre alunos, professores e equipamento didáctico. Essas limitações foram sendo ultrapassadas com recursos a novos meios de comunicação, de que a *Internet* é, neste caso concreto, um óptimo exemplo.

Uma das limitações que se tem mostrado mais difícil de superar é a do acesso remoto dos alunos a laboratórios reais. As várias propostas apresentadas até ao momento implicam normalmente um conjunto de recursos próprios e pouco flexíveis para cada trabalho, com um elevado tempo de aprendizagem para a sua implementação.

Esta dissertação descreve uma metodologia para implementação de laboratórios remotos controlados, via *Internet*, num contexto de ensino à distância para o desenvolvimento de experiências remotas na área de controlo automático. Deste modo qualquer computador com ligação à *Internet* poderá utilizar o simulador para obter valores, parâmetros e dados. O ensino de engenharia na área de controlo automático deve combinar necessariamente teoria e prática. Os estudantes devem obter conhecimentos e desenvolver competências que requeiram o uso intensivo de laboratórios através de equipamentos de grande porte, nomeadamente: robôs industriais, robôs móveis, processos de tanques, motores, etc.

O ensino à distância teve origem ainda no século XIX quando as primeiras Universidades começaram a oferecer cursos por correspondência [Souza, 2001]. A interactividade entre o estudante e o professor era, obviamente, bastante restrita, o que tinha consequências também no número e tipo de cursos que podiam ser oferecidos. Durante a segunda metade do século XX, as novas experiências de ensino à distância usando meios complementares, como por exemplo as emissões televisivas, permitiram melhorar substancialmente o nível de aprendizagem. No entanto, a grande revolução no ensino educativo à distância dar-se-ia apenas nos anos 90 com a introdução do conceito de *e-learning*. Suportado num novo veículo de comunicação, a *Internet* possibilita pela primeira vez a obtenção de um elevado grau de interactividade entre aluno e professor,

aproximando-se da situação real da sala de aula. Apesar disso, o número de opções de cursos continuou restringido a matérias que não requeriam o uso de laboratórios. Nesse sentido, a grande maioria dos cursos de Ciências e Tecnologia, com a excepção óbvia para a Informática, continuavam a não estar disponíveis por esta via.

Esta realidade tem vindo a alterar-se com a difusão do uso de laboratórios remotos. O aumento na largura de banda da transmissão de dados sobre a *Internet* permitiu desenvolver e tornar acessíveis, remotamente, laboratórios reais, com os quais é possível interagir fisicamente. A especificidade da interface de acesso a esses laboratórios e a implementação da necessária interactividade levou a que o seu desenvolvimento se centrasse em torno de montagens específicas. Estes sistemas implicavam novas abordagens na concepção e montagem dos trabalhos, afastando-os das situações reais de laboratório, implicando também elevados custos de projecto e longos períodos de desenvolvimento. Estes condicionalismos tendem a afastar potenciais interessados, limitando as possibilidades de expansão da oferta de cursos já existentes através da plataforma de *e-learning*.

No capítulo dois apresentam-se alguns laboratórios remotos existentes destinados ao ensino.

1.2 Objectivos e Contribuições

O objectivo principal deste trabalho consiste na elaboração de uma arquitectura de um laboratório remoto que permita o estudo das suas características e do seu funcionamento. O desenvolvimento de uma arquitectura deste tipo permite a análise das dinâmicas de sistemas, que poderão ser associados ao desenvolvimento de trabalhos laboratoriais.

Esta dissertação pretende mostrar uma arquitectura de laboratórios remotos baseada em *MATLAB/Simulink* e *LabVIEW* cuja comunicação é feita através do *SIT Connection Manager (Simulation Interface Toolkit)*. Com esse objectivo foi implementado um simulador para supervisão e controlo de sistemas dinâmicos.

1.3 Estrutura da Dissertação

Esta dissertação está dividida em cinco capítulos. No primeiro capítulo é apresentada a motivação para a escolha do tema da dissertação e uma breve descrição histórica sobre o ensino à distância e os objectivos do trabalho que serviu de base à dissertação.

No segundo capítulo, “Laboratórios Remotos”, é feita uma introdução à temática de laboratórios remotos, onde se enumeram algumas vantagens e desvantagens na utilização dos mesmos, apoiado por alguns exemplos de laboratórios remotos já existentes.

O terceiro capítulo, “Arquitectura Proposta e Conceitos Teóricos” consiste na apresentação e descrição da arquitectura proposta. É também composto por uma breve introdução do *Simulation Interface toolkit* que é usado para fazer a comunicação entre o *Simulink* e o *LabVIEW*, das *S-Functions* que são utilizadas para implementar código *MATLAB* no *Simulink* e transformada de Hilbert, utilizada para o cálculo da frequência instantânea do erro de controlo.

No quarto capítulo, “Implementação Prática e Resultados Experimentais” é descrita toda a actividade prática desenvolvida para a realização desta dissertação juntamente com os resultados obtidos.

No quinto e último capítulo apresentam-se as “Conclusões” e apontamentos para trabalho futuro.

Fazem ainda parte desta dissertação dois apêndices. No primeiro apêndice encontra-se a descrição da implementação do simulador do processo PCT-9, enquanto que no segundo apêndice é descrita a implementação relativa ao simulador de processo DTS-200 da Amira.

2 LABORATÓRIOS REMOTOS

Os laboratórios remotos são actualmente alvos de investigação e têm como objectivo de curto prazo a melhoria da segurança, da facilidade, e dos acessos aos laboratórios reais ou virtuais. Com base no trabalho já existente, apresentam-se neste capítulo vários exemplos de laboratórios remotos.

Este capítulo encontra-se dividido em quatro secções. Na primeira secção é explicado o que são laboratórios remotos. Na segunda secção são apresentados alguns desafios dos laboratórios remotos face aos laboratórios tradicionais. Na terceira secção são apresentadas algumas vantagens e desvantagens dos laboratórios remotos. Por fim, na quarta e última secção são apresentados exemplos de software de aquisição de dados para laboratórios remotos e alguns laboratórios remotos existentes.

2.1 Definição de laboratórios remotos

Segundo [Garcia, 2007] os laboratórios remotos permitem que sejam executadas experiências e tarefas laboratoriais através da *Internet*, ou por outro meio de comunicação, sem estarmos ao alcance do equipamento.

Num laboratório tradicional o utilizador interage directamente com o equipamento, podendo ter acções físicas (como carregar em botões, ligar e desligar válvulas, etc.) e podendo também receber respostas visuais, sonoras, etc..

Num laboratório remoto a mesma interacção acontece à distância com a ajuda de uma infra-estrutura remota. Esta infra-estrutura é uma nova camada que se encontra entre o utilizador e o equipamento de laboratório, sendo esta camada responsável por receber as acções do utilizador e devolver a informação do equipamento.

No lado do utilizador, a infra-estrutura desempenha as seguintes funções:

- Fornece uma interface que permite que a experiência seja monitorizada e executada;
- Gere o acesso ao laboratório, de modo a assegurar que só um utilizador possa aceder ao equipamento num determinado tempo.

No lado do equipamento, a infra-estrutura remota desempenha as seguintes funções:

- Monitorização, através de câmaras de vídeo, microfones, sensores, etc.;
- Controlo: através de interfaces I/O, motores, controladores, etc.;
- Assegura que a experiência está pronta a ser novamente utilizada após utilização prévia.

Na Figura 2.1 é apresentado uma possível estrutura de um laboratório remoto.

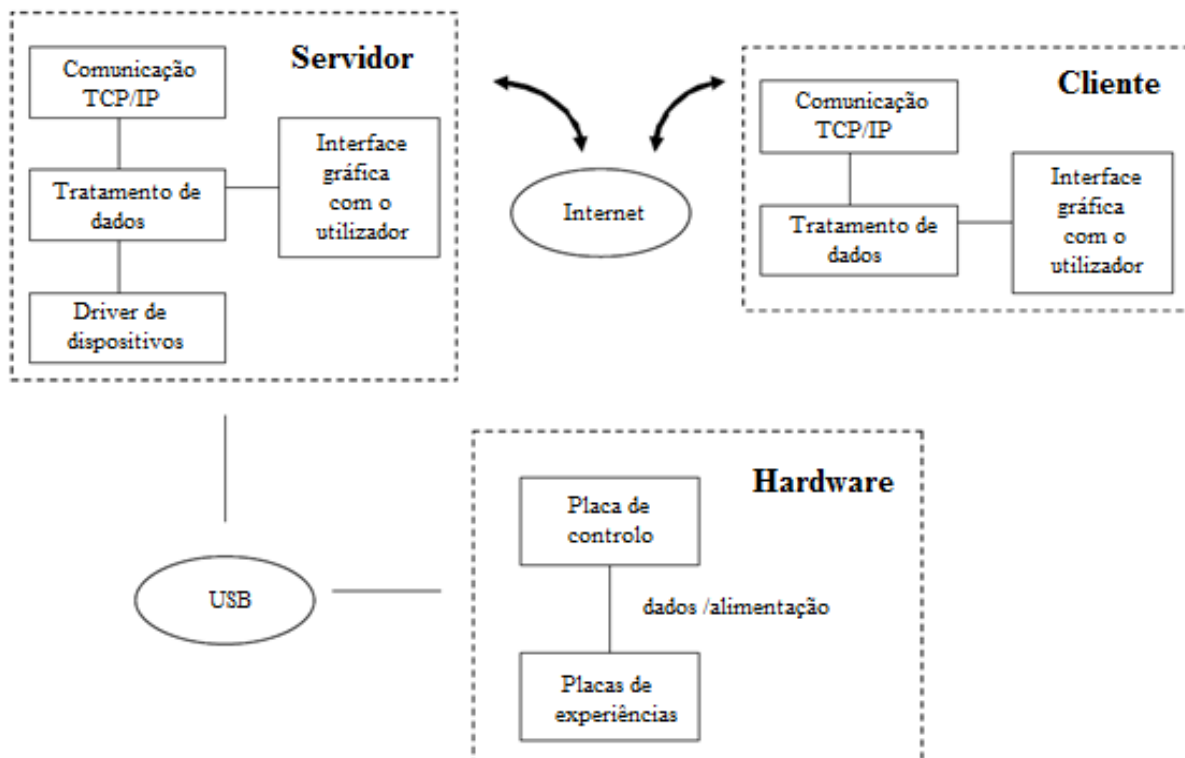


Figura 2.1 - Estrutura de um possível laboratório remoto [Marchezan, 2004].

De entre os vários módulos que compõem esta possível arquitectura, existem três que poderão fazer parte de qualquer tipo de arquitectura:

- O módulo de Comunicação: disponível no Servidor e no Cliente é responsável por estabelecer a comunicação entre o utilizador e o servidor onde está sendo realizada a experiência. Transporta as informações relacionadas com os comandos enviados pelo cliente e os dados enviados pelos instrumentos de medida;
- O módulo Tratamento de Dados: existe tanto no lado do Servidor como no lado do Cliente. A sua função consiste na codificação e decodificação da informação vinda do módulo de comunicação, assim como pelos comandos enviados pelo Cliente através do módulo de Interface Gráfica com o Utilizador;
- O módulo Interface Gráfica com o utilizador tem como funcionalidade permitir a monitorização das operações e o diagnóstico do sistema.

2.2 Desafios face aos laboratórios tradicionais

A substituição de laboratórios tradicionais por laboratórios remotos em procedimentos de pesquisa e didáticos é uma prática consolidada [Eduardo, 2007]. Mesmo assim, ainda existem alguns desafios que os laboratórios remotos têm de enfrentar face aos laboratórios tradicionais:

- **O aumento do número de alunos:** O crescimento e aumento do número de pessoas nas universidades requer um maior uso dos laboratórios.
- **Novos cursos:** O rápido progresso científico e tecnológico resulta em novas áreas de conhecimentos para os estudantes aprenderem. Isto pode requerer a criação de novos cursos e novos requisitos para a aprendizagem prática.
- **Obsolescência:** Os equipamentos modernos, como os computadores, estão a ser constantemente substituídos por tecnologias mais modernas. Isto pode resultar na substituição frequente dos equipamentos de modo a garantir que os alunos se mantenham a par do avanço da tecnologia. A necessidade de actualizações e substituições frequentes podem ser muito dispendiosas.
- **Custos de manutenção:** Os laboratórios exigem frequentes manutenções que são muito dispendiosas.

Um exemplo de um laboratório tradicional é apresentado na Figura 2.2.

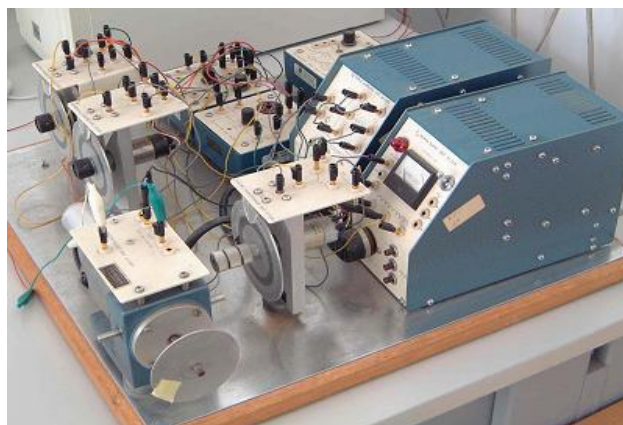


Figura 2.2 - Laboratório tradicional.

2.3 Vantagens e desvantagens

Diversas universidades têm vindo ao longo dos anos a investir no desenvolvimento de laboratórios remotos, de forma a expandirem os seus laboratórios tradicionais. Com o avanço das infra-estruturas tecnológicas, com o desenvolvimento da Web e dos meios multimédia, cada vez mais se recorre a este tipo de laboratórios. Algumas vantagens dos laboratórios remotos face aos laboratórios tradicionais são apresentadas de seguida [Lopes, 2007].

- **Acessibilidade:** Os laboratórios podem ser acedidos 24h por dia, 7 dias por semana, através da *Internet*, sem requerer supervisão;
- **Custos de eficiência:** Face a uma partilha mais eficiente de recursos, as instituições educacionais podem executar experiências laboratoriais com menos equipamentos;
- **Educação à distância:** Cursos que tradicionalmente necessitavam de ser dados em instalações de ensino, podem oferecer a oportunidade do ensino à distância;
- **Consolidação de compromissos:** Os estudantes que trabalham e estudam ao mesmo tempo, podem achar os laboratórios remotos úteis para poderem conciliar melhor os seus compromissos;
- **Gestão:** Através do uso de sistemas de gestão automáticos, a todos os utilizadores dos laboratórios pode ser atribuído um tempo limite de utilização de modo a garantir que todos os utilizadores têm o mesmo tempo de uso do laboratório;

- **Segurança:** Muitas experiências podem originar fogo, choques eléctricos, radiação, etc., o que pode colocar os utilizadores em risco.

Por outro lado, a substituição dos laboratórios tradicionais pode trazer desvantagens para o aluno:

- Não experimenta a real percepção “*hands-on*”;
- Não adquire certas noções de segurança;
- Não interage tão facilmente com elementos de outros grupos;
- Não aprende com os erros;
- As experiências remotas já estão normalmente prontas a serem utilizadas, não incentivando o aluno a reflectir sobre certos parâmetros de preparação.

2.4 Exemplos de software de aquisição e processamento de dados

A aquisição e o envio de dados para a experiência são controlados pelo computador do laboratório através de aplicações. Estes programas podem ser criados usando software específico para controlo, como *LabVIEW* ou *MATLAB*, ou com recurso a linguagens de programação mais genéricas como o C, o C++ ou o VBasic. Será feita de seguida uma breve apresentação de dois softwares de aquisição e tratamento de dados muito utilizados no projecto de laboratórios remotos em instituições de ensino: *LabVIEW* e *MATLAB*.

LabVIEW

O *LabVIEW* é um ambiente de desenvolvimento gráfico da National Instruments que permite, de uma forma rápida e relativamente simples, a criação de aplicações de teste, medição e controlo. Através de uma interface *LabVIEW* é possível interagir com sinais reais, analisar dados e partilhar resultados. Lançado em 1985, originalmente para Macintosh, o *LabVIEW* tem vindo a aumentar o leque de funcionalidades que oferece e, cada vez mais, a permitir que os utilizadores ajustem os sistemas de acordo com as suas necessidades específicas [NI].

MATLAB

O *MATLAB* foi o primeiro e principal produto da MathWorks, que propõe uma gama de produtos de análise e visualização de dados, desenvolvimento de aplicações, simulação, projecto e desenvolvimento de algoritmos [Mathworks].

O *MATLAB* é uma linguagem intuitiva e um ambiente de computação numérico. Apresenta uma série de ferramentas gráficas para visualização e análise de dados, desenvolvimento de algoritmos e aplicações. Combina as vantagens das inúmeras funções matemáticas, estatísticas e gráficas, com uma linguagem de programação de alto nível. É um ambiente largamente utilizado no meio científico e no mundo da engenharia.

2.5 Diferentes abordagens para laboratórios remotos

Na tabela seguinte temos a comparação de diferentes metodologias para a implementação de laboratórios remotos. É de notar que só as principais tecnologias são apresentadas.

Tabela 2.1 - Diferentes abordagens para laboratórios remotos.

| Autores | Servidor | Cliente |
|---|--|-------------------------------|
| [Sánchez, 2004] | <i>MATLAB/Simulink</i> | Navegador Web |
| [Aktan, 1996] | <i>MATLAB/Simulink</i> | Navegador Web |
| [Casini, 2003] | <i>MATLAB/Simulink</i> | Navegador Web |
| [Liou, 1999] | <i>LabVIEW</i> | Navegador Web |
| [Imbrie, 2005] | <i>LabVIEW</i> | Navegador Web e <i>MATLAB</i> |
| [Viedma, 2005] | <i>Java</i> | Navegador Web |
| [Ko, 2001] | NetMeeting | <i>LabVIEW</i> |
| [Bing, 2004] | NetMeeting | <i>LabVIEW</i> |
| [Apkarian, 2000] | <i>MATLAB/Simulink, WinCon e Maple</i> | Navegador Web |
| [Schimid, 2000] | <i>VRML, MATLAB/Simulink, Maple e JAVA</i> | <i>MATLAB</i> |
| [Chan, 2002] | <i>Visual Basic</i> | Navegador Web |
| [Exel, 2000] | <i>WinCon Server</i> | <i>WinCon</i> |
| [Buhler, 2000] | <i>Laboratório real ligado à Internet</i> | Navegador Web |
| [Zeilmann, 2003] | Supervisão com interface OPC | Navegador Web |
| Abordagem proposta nesta dissertação | <i>MATLAB/Simulink</i> | <i>LabVIEW</i> |

3 ARQUITECTURA PROPOSTA E CONCEITOS TEÓRICOS

Este capítulo encontra-se dividido em quatro secções. Na primeira secção é descrita a arquitectura proposta para a realização desta dissertação. Nas seguintes secções são descritos alguns conceitos teóricos usados neste trabalho: na segunda secção é explicado o funcionamento do *SIT Connection Manager*, na terceira secção descreve-se o funcionamento das *S-Functions*, e por fim na quarta secção é explicada a transformada de Hilbert.

3.1 Arquitectura proposta

Existem diferentes tipos de arquitecturas que podem ser utilizadas na construção de um laboratório remoto, como foi demonstrado no capítulo anterior. Após alguma pesquisa sobre qual a arquitectura a usar nesta dissertação, chegou-se à conclusão que seria interessante implementar uma arquitectura conceptual como a que está ilustrada na Figura 3.1.

Esta arquitectura é composta por um bloco Supervisor, por um bloco Rede – TCP/IP e por um bloco Anel de controlo.

O bloco Supervisor irá servir para analisar o desempenho da instalação (detectar oscilações do anel de controlo), analisar o desempenho dos controladores e também para monitorização. Para tal, será possível aceder e alterar os valores dos parâmetros dos controladores ($x=[p_1, p_2, \dots, p_n]$), o valor da referência do anel de controlo e os valores de entrada e de saída da instalação ($i=[\theta_1, \theta_2, \dots, \theta_n]$).

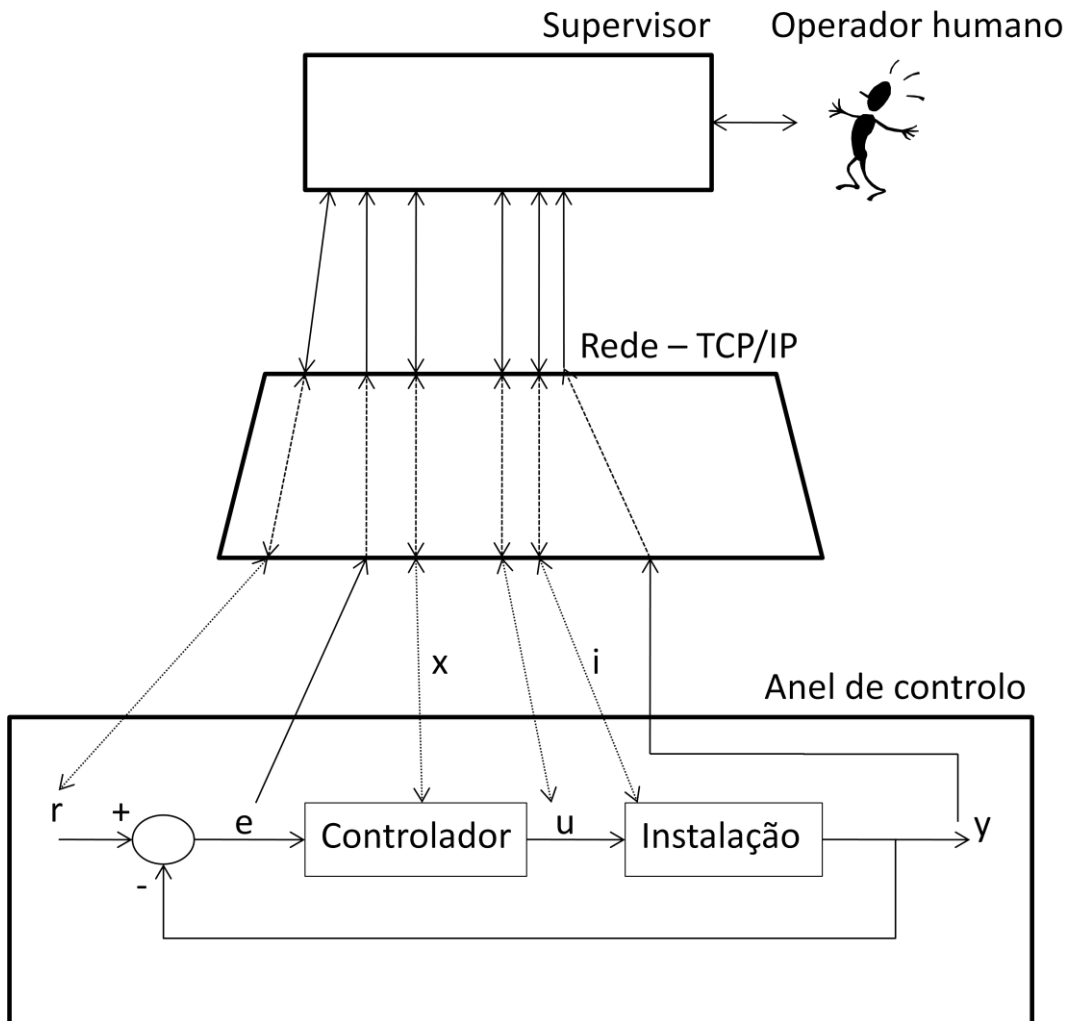


Figura 3.1 - Arquitetura conceptual proposta.

O bloco Rede – TCP/IP será o bloco responsável por toda a comunicação entre o bloco Supervisor e o bloco Anel de controlo.

O bloco Anel de controlo irá conter toda a instalação, bem como os controladores que irão controlar essa mesma instalação.

Para se implementar a arquitetura conceptual ilustrada na Figura 3.1, optou-se por utilizar a arquitetura da Figura 3.2. Nesta arquitetura temos o *LabVIEW* no lado do Supervisor, por este ser um programa que possibilita uma visualização gráfica ao utilizador, e temos o *MATLAB* no lado do Anel de controlo, visto este ser um programa muito utilizado na área de controlo e automação.

Na Figura 3.2 é possível visualizar não só os softwares utilizados por cada computador, mas também como é feita a comunicação entre os dois softwares e entre os próprios softwares.

No computador A, encontra-se implementado o bloco Anel de controlo. Como se pode observar pela Figura 3.2, neste computador todo o código para o controlo dos tanques será implementado em *MATLAB* nas funções *.m*. Estas funções irão comunicar com o *Simulink* através das *S-Functions*, fazendo assim a ligação entre o *MATLAB* e o *Simulink*. A *S-Function* é dividida em segmentos de código que executam uma determinada tarefa, dada a etapa corrente da simulação. O modo de funcionamento das *S-Functions* será descrito mais à frente neste capítulo.

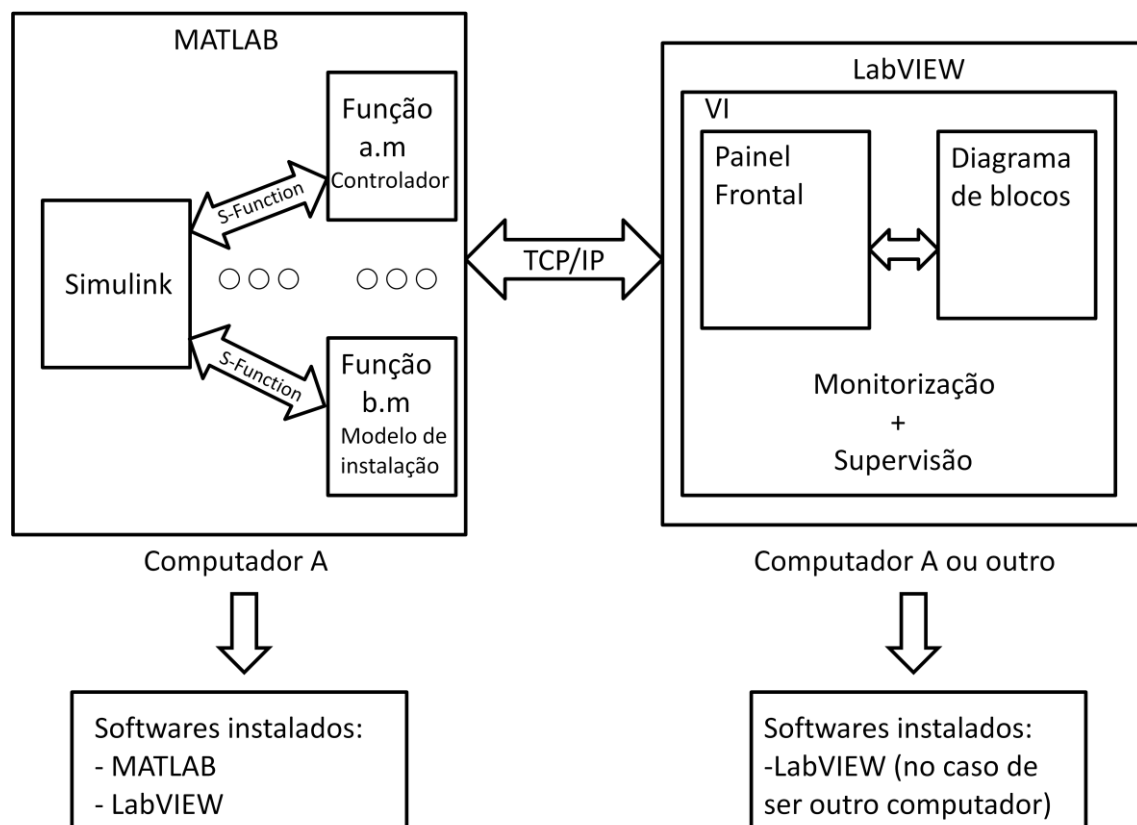


Figura 3.2 - Arquitectura proposta para laboratório remoto com integração do *MATLAB/Simulink* e do *LabVIEW*.

No computador A necessitamos de ter instalado dois softwares: o *MATLAB* e o *LabVIEW*. Apesar de só precisarmos do *MATLAB*, é necessário instalar o *LabVIEW*

porque através do *toolkit* do SIT irá ser criada uma biblioteca no *MATLAB* que possibilitará a comunicação com o *LabVIEW* através do porto 6011.

No lado do “outro computador” encontra-se implementado o bloco Supervisor. Neste computador será feita toda a supervisão e toda a monitorização da instalação. Ao contrário da configuração existente no computador A, só necessitamos de ter instalado o *LabVIEW* com o *toolkit* do SIT.

A comunicação entre os dois computadores é feita através da *Internet*, usando o SIT *Connection Manager*. Esta ferramenta será apresentada e descrita no subcapítulo seguinte.

3.2 Simulation Interface Toolkit (SIT)

O *Simulation Interface Toolkit* (SIT) fornece métodos para criar uma interface de utilizador do *LabVIEW* para um modelo *Simulink*, convertendo um modelo *Simulink* numa biblioteca de vínculo dinâmico (DLL), e executando um modelo de simulação de um alvo em tempo real, combinando os recursos *Simulink* e *Real-Time Workshop* com o *LabVIEW*. O *Simulation Interface Toolkit* ajuda na importação de modelos de simulação em *LabVIEW*.

Segundo [NI], o *LabVIEW*, o *Simulink*, e o *Real-Time Workshop* são linguagens de programação gráficas que utilizam ícones em vez de linhas de texto para criar aplicações. As aplicações do *LabVIEW* são chamadas de instrumentos virtuais ou *Vis*, porque a sua aparência e o seu funcionamento, muitas vezes imitam instrumentos físicos, tais como multímetros e osciloscópios.

O *Simulink*, um *add-on toolkit* para o *MATLAB*, oferece um ambiente gráfico para a concepção e execução interactiva de modelos de sistemas dinâmicos.

Para criar um *LabVIEW User Interface* para um modelo em *Simulink*, o SIT permite criar uma interface de utilizador no *LabVIEW* que pode ser usada para interagir com um modelo em *Simulink*. Com esta interface de utilizador, é possível manipular os parâmetros do modelo e visualizar os dados de saída do modelo *Simulink*.

Para utilizar o SIT, o *LabVIEW* e o *MATLAB* deverão estar em execução no computador *host*, caso não haja outro computador. O Computador *host* é o computador onde pode ser criada a interface do utilizador do *LabVIEW*. O computador *host* deverá ser um PC com o sistema operativo *Windows*. O intercâmbio de dados do *LabVIEW* com o *MATLAB* e *Simulink* pode ser feito através de TCP/IP.

Para o *LabVIEW* comunicar com sucesso com o *MATLAB* e o *Simulink*, o *MATLAB* deve estar a correr. Quando o *MATLAB* é iniciado, o SIT também é iniciado automaticamente, o que permite a comunicação do *LabVIEW* e do *MATLAB*.

A Figura 3.3 ilustra a interacção entre o *LabVIEW* e o *MATLAB/Simulink*.

No computador *host* é criado um *host VI*, que é um painel frontal *LabVIEW*, que fornece a interface do utilizador com o modelo *Simulink*. Este painel frontal contém controladores e indicadores que correspondem aos parâmetros *Simulink*, os quais se deseja visualizar ou modificar. Os valores dos controladores podem ser alterados e é possível visualizar imediatamente a mudança nos indicadores.

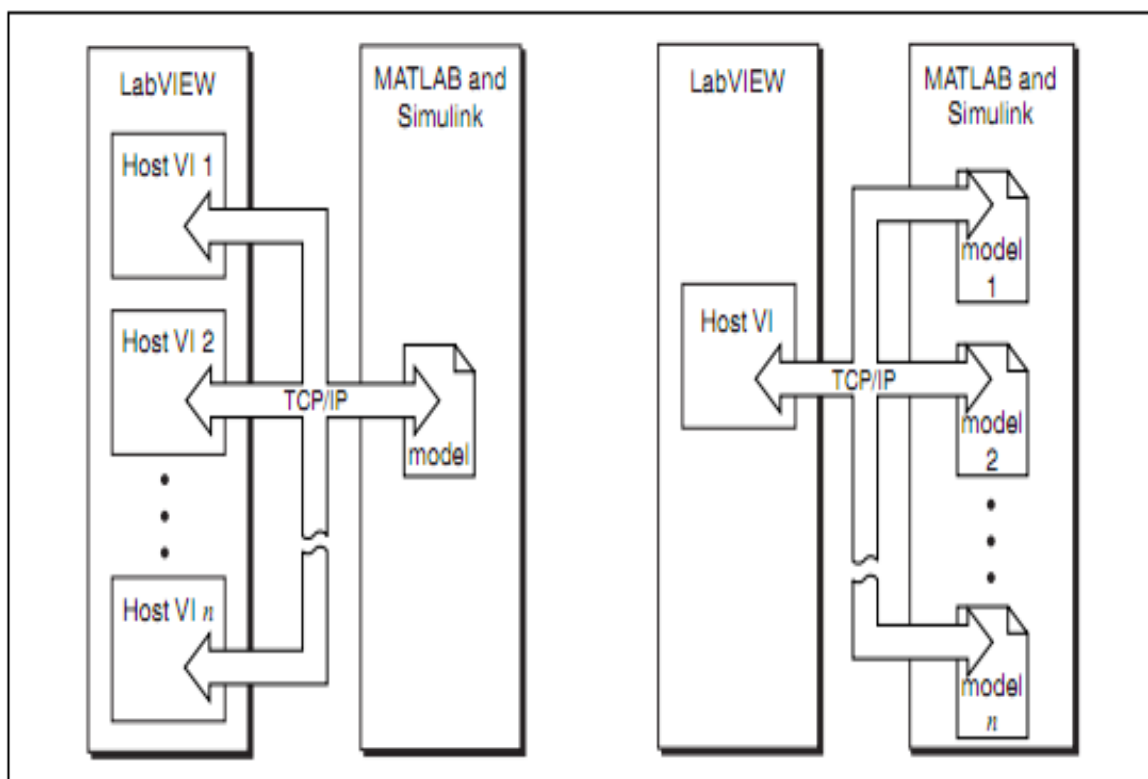


Figura 3.3 - Ligação de várias *Vis* e de vários modelos *Simulink*. [Mathworks].

Utilizando o *SIT Connection Manager*, é possível especificar a relação entre os controladores e indicadores do *LabVIEW* com os parâmetros do *Simulink*. Depois de configurar o *SIT Connection Manager*, este gera automaticamente o código do diagrama de blocos necessário para estabelecer as relações entre o VI do *LabVIEW* e o modelo *Simulink*.

Através do SIT podem ser ligados vários *Vis* criados no computador *host* para o mesmo modelo *Simulink*. É possível também ligar vários modelos *Simulink* no mesmo computador com um único VI criado no computador *host*. Por outro lado, se os modelos *Simulink* estão em computadores diferentes, não poderá ser feita a ligação para o VI.

Na Figura 3.3 podem ser observados os dois modos de conectar o *LabVIEW* com o *Simulink*. Primeiramente é necessário configurar o modelo em *Simulink* para se comunicar com o *LabVIEW*, sendo depois criado um *LabVIEW* VI *host* que automaticamente chama, interage e corre o modelo em *Simulink*.

Configuração do modelo em *Simulink*

Para que seja estabelecida comunicação entre o *LabVIEW* e o modelo em *Simulink*, este deve ser configurado correctamente. Ou seja, a título de exemplo para um sistema com tanques de água:

1. Salve os ficheiros a.mdl e b.vi no computador.
2. Execute o *MATLAB* e verifique se na janela de comando do *MATLAB* aparece a seguinte mensagem:

Iniciando o Servidor SIT no porto 6011

SIT Server iniciado

3. Digite *Simulink* na janela de comando do *MATLAB* para o lançamento do *Simulink*.
4. Seleccione **Ficheiro, Abrir** e seleccione o a.mdl que é o ficheiro que contém o modelo em *Simulink*.
5. No modelo de simulação, em cada ponto que se pretenda monitorizar, é necessário adicionar um SI SIT Block ao modelo em *Simulink* para permitir que o *LabVIEW* consiga ler os dados desse modelo.

6. Na janela **Biblioteca** carregue sobre **NI SIT Blocks** e coloque um **Signal Probe** no modelo.
7. Salve o modelo em *Simulink*.

A título de exemplo, é apresentada na Figura 3.4 um possível modelo de uma instalação de um laboratório.

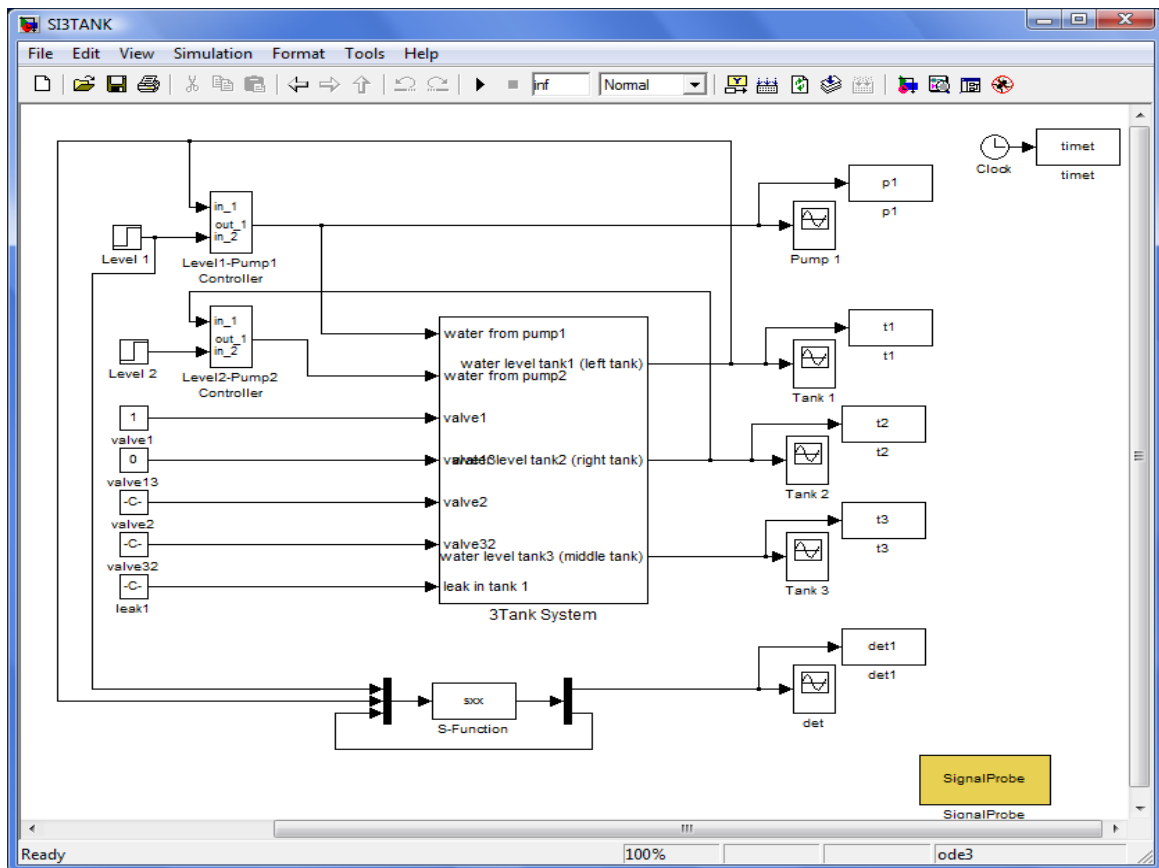


Figura 3.4 - Modelo em *Simulink*.

Configuração da VI em *LabVIEW*

Para interagir com o modelo dos tanques usando uma interface de utilizador do *LabVIEW*, é necessário previamente criar a mesma interface. Esta pode ser criada através da seguinte sequência de comandos:

1. Execute o *LabVIEW* e clique em **NEW, Bani VI**.

2. Carregue com o botão direito do rato no painel principal, clique em **DSC Module – Vessels – Tank** e coloque três tanques no painel principal.
3. Identifique os tanques e altere a escala para 0-0,6.
4. Carregue novamente no botão direito, vá a **Numeric Controls – Knob** para indicar o fluxo de água que sai da bomba de água de enchimento dos tanques.
5. Identifique os indicadores Pump In Tank1 e Pump In Tank2.
6. Vá a **Graph Indicators – Chart** e coloque cinco no painel principal. O painel VI deverá ter uma configuração semelhante à Figura 3.5:

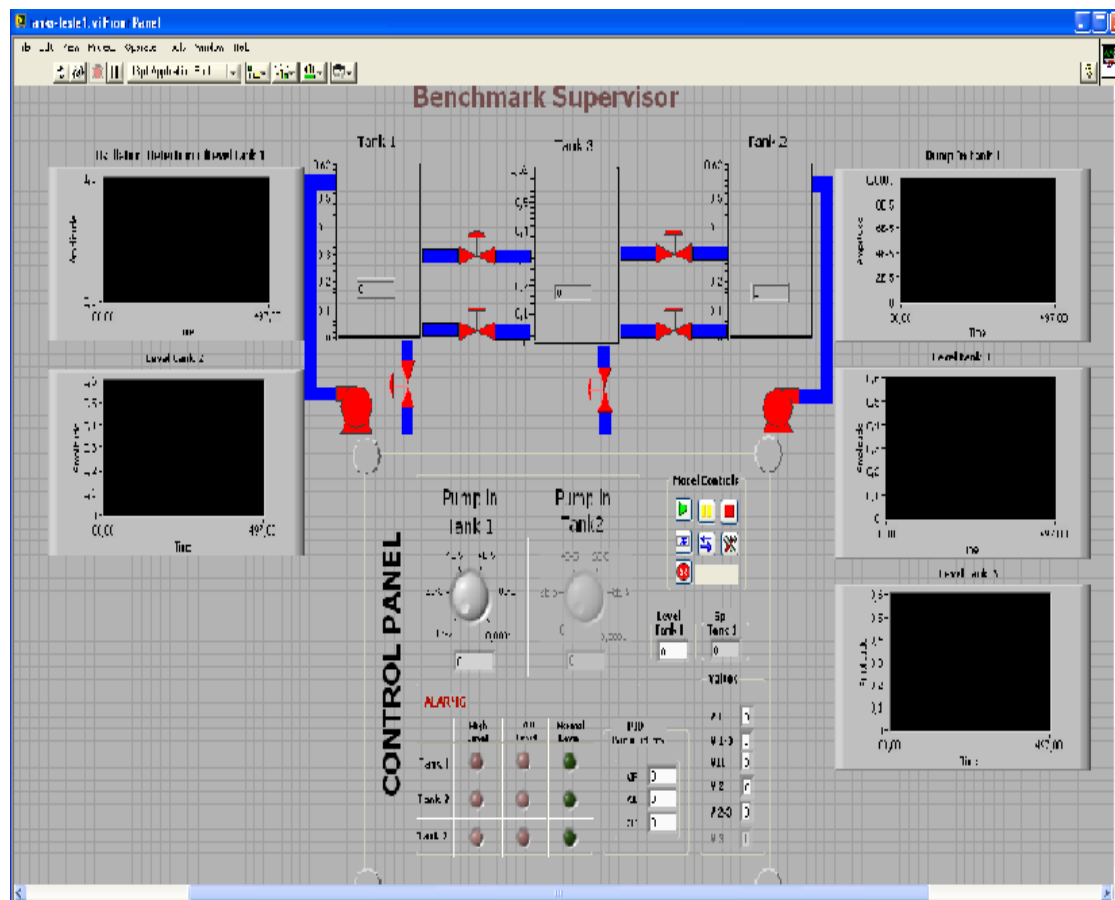


Figura 3.5 - Painel frontal do *LabVIEW*.

Ligações entre *LabVIEW* e *Simulink*

O SIT permite ao *LabVIEW* comunicar e correr o modelo em *Simulink*, desde que as ligações estejam feitas correctamente. O *MATLAB* e o *Simulink* devem permanecer abertos durante a configuração das ligações.

1. No menu do painel frontal, selecione **Tools – SIT Connection Manager**.
2. No **Execution Host**, selecione **Simulation Environment** e no **Simulation IP Address** coloque *LocalHost*, no caso do modelo em *Simulink* estar no mesmo computador que o VI do *LabVIEW*, ou coloque o IP do computador onde tem o modelo em *Simulink*.
3. Em **Current Model** coloque o caminho para o ficheiro .mdl criado antes e em **Project Directory** coloque a directoria onde quer que o projecto corra.
4. De seguida vá a **Category** e carregue em **Mappings** e verá os controladores e os indicadores. Carregue em cada um e selecione a que parte do modelo em *Simulink* a que cada um pertence. Por exemplo **Indicador – Tank1** liga a **si3tank\3Tank System: 1**. A janela SIT Connection Manager deverá ter a configuração apresentada na Figura 3.6.
5. Clique no botão **OK**.

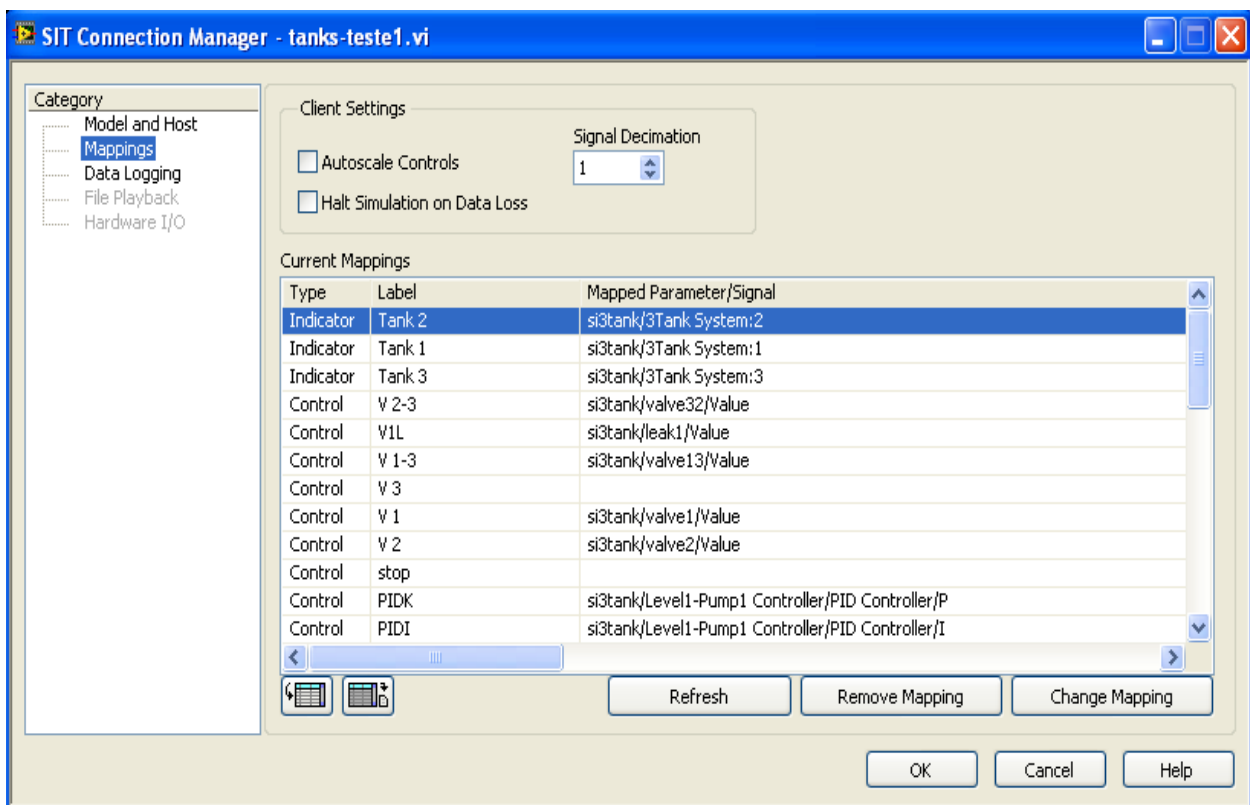


Figura 3.6 - SIT Connection Manager.

Note-se que o SIT gera um diagrama de blocos de código que rege a interligação entre o painel frontal do *LabVIEW* e o modelo em *Simulink*. Aparece no painel frontal um novo

conjunto de botões (*Model Controls*); esses botões são gerados pelo *SIT Connection Manager*.

Clique no botão **Run** para executar o VI.

Para executar o modelo em *Simulink*, deverá carregar no botão **Run Simulation** do **Model Controls**.

3.3 Funções “S” para integração *MATLAB/Simulink*

Quando se pretende executar código *MATLAB* no ambiente *Simulink*, podemos chamar funções programadas em *MATLAB* para analisar dados ou para visualização. Tais funções podem ser executadas através das Funções “S” (*S-Functions*). Além disso, o *Simulink* permite a utilização do código *MATLAB* para projectar algoritmos embutidos que podem ser implementados por meio de código com o resto do modelo em *Simulink*. É também possível incorporar código escrito à mão directamente num modelo, permitindo a criação de blocos personalizados.

3.3.1 Funções “S” (*S-Functions*)

Segundo [Mathworks] Uma Função “S” é um bloco do *Simulink* numa linguagem de programação, que pode ser codificada em *MATLAB*, C / C++, *Fortran* ou *Ada*. Na Figura 3.7 encontra-se ilustrada a representação de um bloco de uma Função “S” do *MATLAB/Simulink*.



Figura 3.7 - Bloco Função “S”.

Na janela de parâmetros do bloco da Função “S”, como ilustra a Figura 3.8, deve ser indicado o nome da função. Se for codificada em *MATLAB*, o código deve ser guardado numa m-file com o mesmo nome da função.

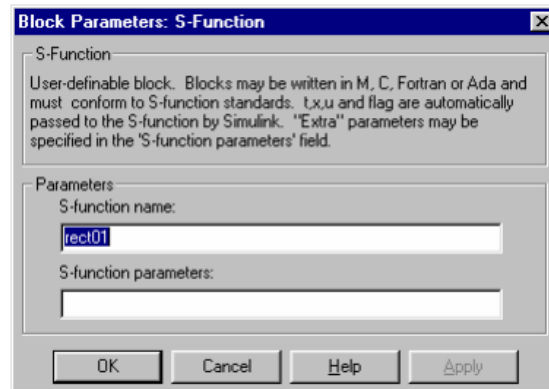


Figura 3.8 - Parâmetros do bloco Função “S”.

Um bloco do *Simulink* é representado na Figura 3.9.

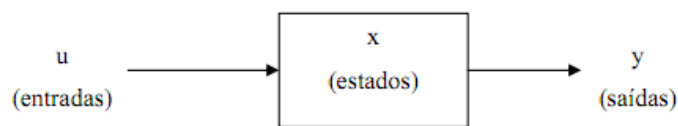


Figura 3.9 - Representação de um bloco do *Simulink*.

De acordo com um modelo de espaço de estados, e sabendo que se tem estados contínuos e discretos, da forma

$$x = x_d + x_c \quad (1)$$

obtêm-se as seguintes equações relativas à derivada dos estados contínuos e à equação de saída, respectivamente

$$\dot{x}_c = Ax_c + Bu \quad (2)$$

$$y = Cx + Du \quad (3)$$

Onde as matrizes A, B, C e D são chamadas de matrizes de estado do sistema; $u(t)$ é o sinal de entrada do sistema; $x(t)$ é o vector de variáveis de estado e $y(t)$ é a saída do sistema dinâmico.

A codificação em *MATLAB* toma a forma de uma função com o seguinte aspecto:

```
[sys,x0,str,ts] = f (t, x, u, flag, p1, p2, ...)
```

Os parâmetros de entrada da respectiva função estão descritos na Tabela 3.1.

Tabela 3.1 - Parâmetros de entrada de uma Função “S”.

| | |
|-------------|--|
| f | Nome da Função “S” |
| t | Tempo corrente |
| x | Vector de estado |
| u | Entrada do bloco |
| flag | Indica uma tarefa a ser desempenhada durante a simulação |
| p1, p2, ... | Parâmetros adicionais |

Quando o tempo de amostragem varia normalmente p1 é utilizado para indicar esse valor à Função “S”. Os valores destes parâmetros devem ser indicados na janela de parâmetros do bloco, mostrada na Figura 3.8, na linha de entrada intitulada *S-function parameters* separados por vírgulas. Podem ser constantes ou variáveis definidas no *workspace* do *MATLAB*.

Os parâmetros de saída da função apresentada anteriormente são descritos na Tabela 3.2.

Tabela 3.2 - Parâmetros de saída de uma Função “S”.

| | |
|-----|--|
| sys | Argumento de retorno genérico. Os valores dependem de <i>flag</i> . (ex. se <i>flag</i> = 3, sys contém a saída da Função “S” (ou do bloco)) |
| x0 | Valores iniciais dos estados |
| str | Reservado para uso futuro. Funções “S” devem retornar a matriz vazia []. |
| ts | Matriz com duas colunas contendo o intervalo de amostragem |

A simulação segue o diagrama da Figura 3.10. No início todos os blocos são inicializados (*flag*=0). Inicia-se então um ciclo de simulação onde em primeiro lugar é calculado o próximo instante de amostragem (*flag*=4), de modo que o cálculo da saída (*flag*=3) seja executado apenas após a amostragem de novos dados. Seguidamente são

actualizados os estados discretos ($flag=2$). Antes de se aguardar pela próxima amostra, são ainda calculadas as derivadas dos estados contínuos ($flag=1$).

Podem ainda ser efectuadas algumas tarefas antes do fim da simulação ($flag=9$).

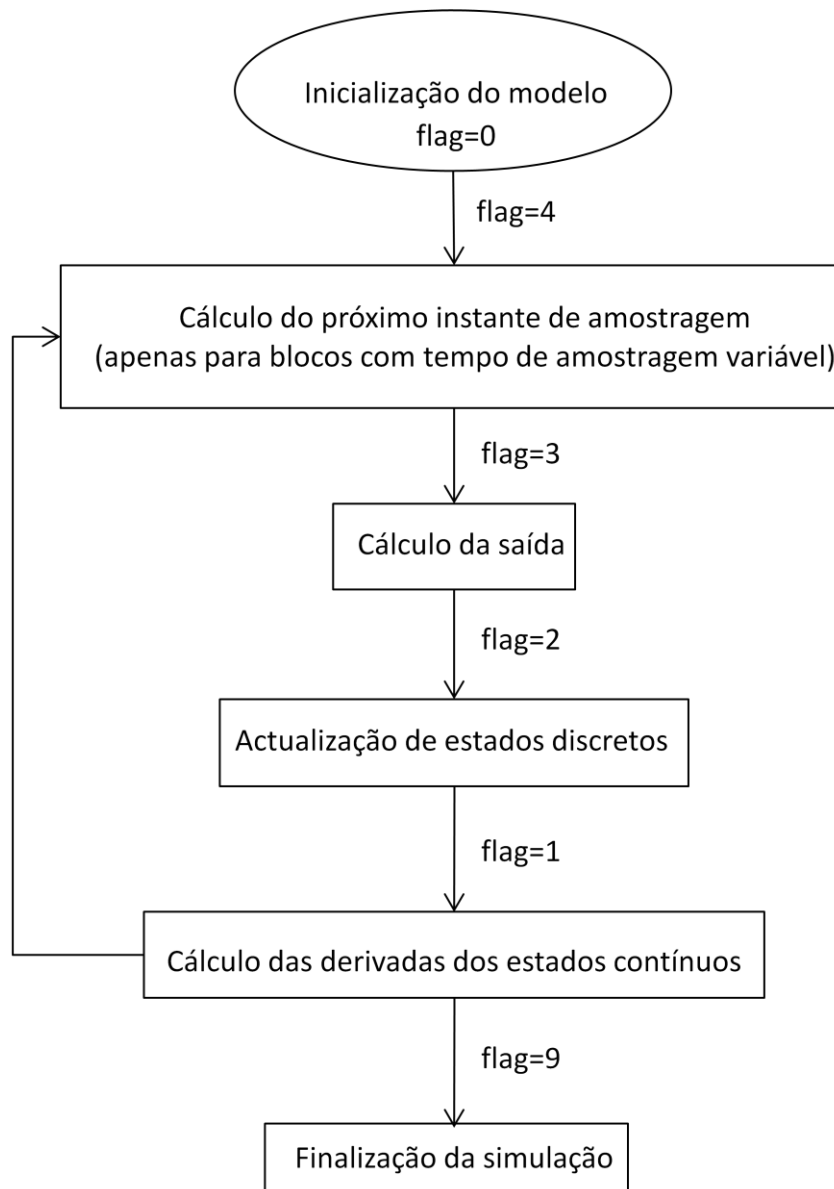


Figura 3.10 - Etapas de uma simulação e correspondência com o valor de $flag$.

Como ilustra a Figura 3.10, as tarefas que são efectuadas durante a simulação dependem do valor do parâmetro de entrada $flag$. Este parâmetro é fornecido pelo próprio

Simulink, cada vez que chama a Função “S” de modo a indicar qual a etapa em que a simulação se encontra.

3.3.2 Implementação de Funções “S”

Neste ponto são apresentados alguns exemplos de Funções “S”. Basicamente uma Função “S” é dividida em segmentos de código que executam uma determinada tarefa, dada a etapa corrente da simulação, e que é especificada pelo valor de *flag*:

```
[sys, x0, str, ts] = sfunc (t, x, u, flag)
```

```
if flag==0  
(...)
```

```
elseif flag == 4  
(...)
```

- **Entrada e saída únicas**

No exemplo da Figura 3.11 admite-se que o tempo de amostragem é variável e portanto é passado como primeiro parâmetro adicional, *ts*. Deste modo o valor deste parâmetro deve ser especificado na janela de parâmetros do bloco. Este exemplo consiste simplesmente num ganho que duplica o valor do sinal de entrada:



Figura 3.11- Diagrama para a Função “S” *gain01*.

O algoritmo referente à Função “S” *gain01* é apresentado de seguida:

```
function [sys, x0, str, ts] = gain01(t, x, u, flag, ts)
```



```

%inicialização

if flag == 0

    %estados contínuos = 0

    %estados discretos = 0

    %saídas = 1

    %entradas = 1

    %direct feedthrough deve ser 1 se u usado em flag=3

    %sample times = 1

    sys = [0 0 1 1 0 1 1];

    x0 = [ ];

    str = [ ];

    ts = [-2 0];          %tempo de amostragem variável

    %Calcula próximo instante de amostragem

elseif flag == 4

    ns = t / ts;          %ns n° de amostras

    sys = (1 + floor(ns + 1e-13*(1+ns)))*ts; %instante da próxima amostra

    %Calcula a saída como o dobro da entrada

elseif flag == 3

    sys = 2*u(1);

    %default

else

    sys = [ ]; %não faz nada

end

```

- **Entradas e saídas múltiplas**

Uma função pode receber mais que uma entrada. Se na segunda entrada se encontrar o factor ganho, o diagrama de blocos respectivo é o da Figura 3.12.

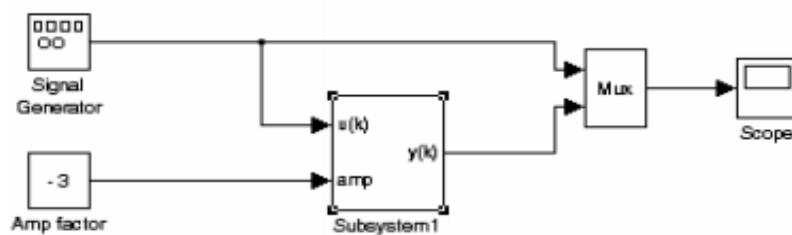


Figura 3.12 - Ganho variável.

Repare-se que a Função “S” juntamente com as suas entradas e saídas foram agrupadas num subsistema. No caso de múltiplas entradas é utilizado um *Multiplexer* de modo a agrupar os sinais independentes. Quanto à saída consiste num vector com tantas posições como o número de saídas de modo que se for ligada a um *Scope* apresenta tantos sinais como o número de saídas.

As alterações que devem ser efectuadas ao algoritmo anterior são:

```
else if flag == 0 %Inicialização
sys = [ 0 0 1 2 0 1 1 ]
elseif flag == 3 %Calcula a saída como um factor da 1ª entrada dado pela 2ª
sys = u(2)*u(1);
```

- **Memorização de estados da S-function entre iterações**

Nos casos anteriores não foram necessários estados. Vamos considerar a hipótese de ser necessário guardar um valor da etapa anterior para efectuar um qualquer cálculo. Uma forma de concretizar esta hipótese (*the hard way*), consiste em guardar esse valor num estado discreto.

O exemplo da Figura 3.13 calcula a diferença entre a entrada corrente $u(k)$ e a entrada anterior $u(k-1)$. Para isso deve ser memorizado o valor da entrada anterior.

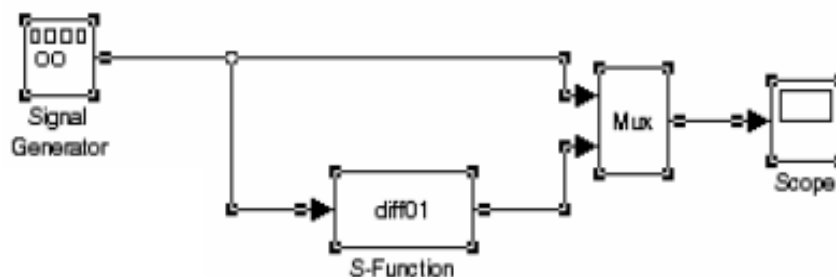


Figura 3.13 - Diferença entre a entrada actual e a anterior.

3.4 Transformada de Hilbert

A transformada de Hilbert é usada neste trabalho para calcular a frequência instantânea do erro de controlo. O cálculo da transformada de Hilbert é feito pelo *MATLAB* usando a função *hilbert()*.

A transformada de Hilbert é uma ferramenta matemática que pode ser usada para descrever um sinal analítico, a partir de um sinal puramente real.

A definição de frequência, que se define pelo inverso do período, para que se possa fazer sentido, exige que o sinal que se esteja a estudar tenha pelo menos a duração de um período. A definição de frequência instantânea pretende estender a definição temporal limitada, a cada ponto (frequência instantânea).

Segundo [Pinto, 2009] e [Liu, 2006] existem duas razões para que a noção de frequência instantânea seja mal aceite/compreendida. A primeira razão é histórica. Segundo a tradição de análise de sinal por via de transformada de Fourier, o sinal é decomposto em somas infinitas de sinusóides com amplitude constante, das quais se extrai a frequência com referência ao período de funções base. Sendo necessário um tempo mínimo, pode parecer estranho aceitar que a frequência possa ser definida pontualmente. A segunda prende-se com a não unicidade na forma de se definir frequência instantânea.

Para um dado sinal $s(t)$, a transformada de Hilbert do sinal s define-se por:

$$H(t) = y(t) = \frac{1}{\pi} P \int_{-\infty}^{+\infty} \frac{s(\tau)}{t-\tau} d\tau \quad (4)$$

Onde P representa o valor principal de Cauchy. Com esta definição estabelece-se de que forma a partir de $s(t)$ se determina $y(t)$ que permite obter o sinal $z(t)$:

$$z(t) = s(t) + iy(t) = a(t)e^{i\theta(t)} \quad (5)$$

Onde

$$a(t) = \sqrt{s(t)^2 + y(t)^2}, \quad \theta(t) = \arctan\left(\frac{y(t)}{s(t)}\right).$$

Apesar de existirem outras hipóteses de se obter a parte imaginária de $z(t)$, a transformada de Hilbert permite obter desta forma um sinal analítico (não apresenta componentes de frequências negativas). Usando-se a equação **5** para a obtenção do sinal analítico, pode definir-se a frequência instantânea por:

$$\omega = \frac{d\theta(t)}{dt}. \quad (6)$$

Nas equações anteriores, a amplitude e a frequência instantânea são funções de tempo.

A definição de frequência instantânea na equação **6** continua a ser a que é mais bem aceite e a que é mais consistente e melhor se adequa a dados não estacionários. Pelo facto de se usar o método da fase estacionária como justificação para se obter a equação **5**, tal aproximação continua a ser tida como uma das limitações para o seu uso.

4 IMPLEMENTAÇÃO PRÁTICA E RESULTADOS EXPERIMENTAIS

Uma vez apresentada a arquitectura proposta e os conceitos teóricos, são apresentados neste capítulo os simuladores que foram feitos ao longo desta dissertação. Devido a contratempos que foram aparecendo ao longo do decorrer desta dissertação, foram necessários fazer três simuladores de modo a conseguir obter o laboratório remoto desejado. Neste capítulo vamos nos focar no simulador para o modelo dos três tanques, os outros dois simuladores são apresentados nos Apêndice A e B.

Este capítulo encontra-se dividido em três secções. Na primeira secção é descrito o processo dinâmico (“*three-tank benchmark*”). Na segunda é descrito o simulador para o modelo dos três tanques. Na terceira encontram-se os resultados experimentais obtidos com simulações feitas com o simulador do modelo dos três tanques.

4.1 Processo dinâmico (“*three-tank benchmark*”)

O sistema dinâmico em estudo é composto por três tanques associados, como está representado na Figura 4.1. Estes tanques são ligados por tubos que podem ser controlados por diversas válvulas. A água entra para os tanques direito e esquerdo através de duas bombas idênticas. Para o tanque T3 os valores qualitativos para os diferentes níveis de água são: *baixo* = $[0 \dots 9] \text{ cm}$, *médio* = $[9 \dots 11] \text{ cm}$, *alto* = $[11 \dots 60] \text{ cm}$.

Para o presente trabalho só são utilizados o tanque da esquerda (T1) e o do meio (T3). O tanque da direita (T2) e a bomba P2 não são usados, passando a ser hardware redundante. O objectivo principal dos dois tanques usados é fornecer uma quantidade contínua de fluxo de água Q_N ao consumidor. O nível de água no tanque do meio (T3) tem de ser mantido a um nível $h_3 = \textit{médio}$. O tanque 1 é abastecido através da bomba P1 até ao valor nominal $h_1 = 50 \text{ cm}$. A água que corre entre os dois tanques pode ser controlada pelas válvulas V_1 e V_{13} . Estas válvulas só podem estar totalmente abertas ou totalmente fechadas. A válvula V_1 é usada para controlar o nível de água no tanque T3 e

encontra-se a uma altura de 30 cm. A válvula V_{1L} que pode ser usada para simular uma fuga no tanque T1, está fechada. As válvulas V_{13} e V_{32} encontram-se rente ao fundo dos tanques.

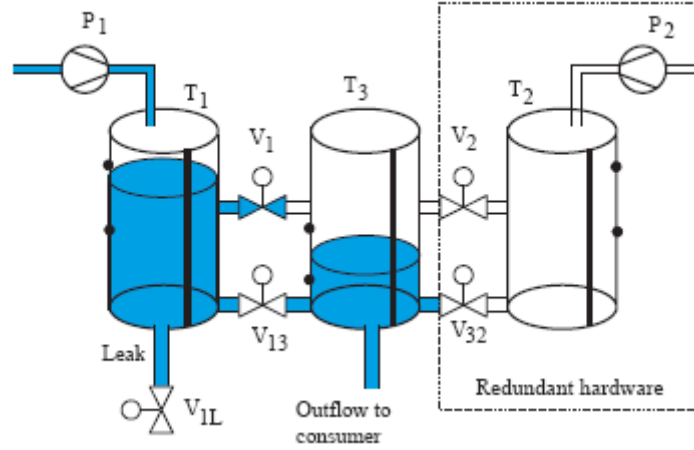


Figura 4.1 - Sistema de três tanques [Heiming, 2006].

4.1.1 Modelo dinâmico do sistema

Do ponto de vista teórico, o sistema acoplado dos três tanques é um típico sistema híbrido. Dependendo do nível de água e da posição das válvulas, são válidos diferentes modelos de estados não lineares. Em geral, o fluxo de água Q_{ij} que vai do tanque i para o tanque j pode ser calculado usando a lei de *Toricelli*

$$Q_{ij} = az \times S \times \text{sgn}(h_i(t) - h_j(t)) \times \sqrt{2 \times g \times |h_i - h_j|} \quad (7)$$

em que:

- az é o termo de correlação de fluxo
- S é a área da secção da válvula de conexão
- g é a constante de gravidade
- h_i, h_j os níveis de água dos dois tanques (i e j).

A variação do volume de água V no tanque, pode ser calculada através da seguinte fórmula:

$$\dot{V} = A \times \dot{h}(t) = \sum Q_{in} - \sum Q_{out} \quad (8)$$

em que:

- A é a área da secção do tanque cilíndrico
- h é a altura de água no tanque

$\sum Q_{in}$ é a soma de toda a quantidade de água que entra no tanque e $\sum Q_{out}$ é a soma de toda a quantidade de água que sai do tanque.

Em geral, as equações não lineares para os tanques são:

$$\dot{h}_1 = \frac{Q_1^{P1} - Q_{13}^{V1} - Q_{13}^{V13} - Q_{fuga}^{V1L}}{A} \quad (9)$$

$$\dot{h}_2 = \frac{Q_2^{P2} - Q_{23}^{V2} - Q_{23}^{V32}}{A} \quad (10)$$

$$\dot{h}_3 = \frac{Q_{13}^{V1} + Q_{13}^{V13} + Q_{23}^{V2} + Q_{23}^{V32} - Q_N}{A} \quad (11)$$

em que o fluxo Q_{ij} depende dos níveis h_1 , h_2 e h_3 , bem como da posição das válvulas. Dependendo do nível de água $h_i < 30 \text{ cm}$ ou $h_i > 30 \text{ cm}$ existem oito modos de operação diferentes do sistema:

- 1- $h_1 < 30 \text{ cm}; \quad h_2 < 30 \text{ cm}; \quad h_3 < 30 \text{ cm}$

$$Q_{13}^{V1} = 0$$

$$Q_{23}^{V2} = 0$$

- 2- $h_1 < 30 \text{ cm}; \quad h_2 < 30 \text{ cm}; \quad h_3 \geq 30 \text{ cm}$

$$Q_{13}^{V1} = az \times S \times \text{sgn}(0,3 - h_3) \times \sqrt{2 \times g \times |0,3 - h_3|} \times \text{Pos}(V_1)$$

$$Q_{23}^{V2} = az \times S \times \text{sgn}(0,3 - h_3) \times \sqrt{2 \times g \times |0,3 - h_3|} \times \text{Pos}(V_2)$$

- 3- $h_1 < 30 \text{ cm}; \quad h_2 \geq 30 \text{ cm}; \quad h_3 < 30 \text{ cm}$

$$Q_{13}^{V1} = 0$$

$$Q_{23}^{V2} = az \times S \times \text{sgn}(h_2 - 0,3) \times \sqrt{2 \times g \times |h_2 - 0,3|} \times \text{Pos}(V_2)$$

- 4- $h_1 < 30 \text{ cm}; \quad h_2 \geq 30 \text{ cm}; \quad h_3 \geq 30 \text{ cm}$

$$Q_{13}^{V1} = az \times S \times \text{sgn}(0,3 - h_3) \times \sqrt{2 \times g \times |0,3 - h_3|} \times \text{Pos}(V_1)$$

$$Q_{23}^{V2} = az \times S \times \text{sgn}(h_2 - h_3) \times \sqrt{2 \times g \times |h_2 - h_3|} \times \text{Pos}(V_2)$$

$$5- \quad h_1 \geq 30cm; \quad h_2 < 30cm; \quad h_3 < 30cm$$

$$Q_{13}^{V1} = az \times S \times sgn(h_1 - 0,3) \times \sqrt{2 \times g \times |h_1 - 0,3|} \times Pos(V_1)$$

$$Q_{23}^{V2} = 0$$

$$6- \quad h_1 \geq 30cm; \quad h_2 < 30cm; \quad h_3 \geq 30cm$$

$$Q_{13}^{V1} = az \times S \times sgn(h_1 - h_3) \times \sqrt{2 \times g \times |h_1 - h_3|} \times Pos(V_1)$$

$$Q_{23}^{V2} = az \times S \times sgn(0,3 - h_3) \times \sqrt{2 \times g \times |0,3 - h_3|} \times Pos(V_2)$$

$$7- \quad h_1 \geq 30cm; \quad h_2 \geq 30cm; \quad h_3 < 30cm$$

$$Q_{13}^{V1} = az \times S \times sgn(h_1 - 0,3) \times \sqrt{2 \times g \times |h_1 - 0,3|} \times Pos(V_1)$$

$$Q_{23}^{V2} = az \times S \times sgn(h_2 - 0,3) \times \sqrt{2 \times g \times |h_2 - 0,3|} \times Pos(V_2)$$

$$8- \quad h_1 \geq 30cm; \quad h_2 \geq 30cm; \quad h_3 \geq 30cm$$

$$Q_{13}^{V1} = az \times S \times sgn(h_1 - h_3) \times \sqrt{2 \times g \times |h_1 - h_3|} \times Pos(V_1)$$

$$Q_{23}^{V2} = az \times S \times sgn(h_2 - h_3) \times \sqrt{2 \times g \times |h_2 - h_3|} \times Pos(V_2)$$

Para os oito casos, podem ser utilizadas as seguintes equações:

$$Q_{13}^{V1} = az \times S \times sgn(h_1 - h_3) \times \sqrt{2 \times g \times |h_1 - h_3|} \times Pos(V_1) \quad (12)$$

$$Q_{23}^{V2} = az \times S \times sgn(h_2 - h_3) \times \sqrt{2 \times g \times |h_2 - h_3|} \times Pos(V_2) \quad (13)$$

$$Q_N = az \times S \times \sqrt{2 \times g \times h_3} \quad (14)$$

$$Q_{V1L}^{fuga} = az \times S \times \sqrt{2 \times g \times h_1} \times Pos(V_{1L}) \quad (15)$$

A posição das válvulas $Pos(.)$, pode ser 0 se a válvula estiver fechada ou 1 se a válvula estiver aberta. A altura máxima de cada tanque é de 60cm e os parâmetros são:

- $az = 1$
- $A = 0,0154 \, m^2$
- $g = 9,81 \, m/s^2$
- $S = 3,5 \times 10^{-5} \, m^2$
- $Q_{Max} = 0,1 \times 10^{-3} \, m^3/s$

4.1.2 O problema do *benchmark*

Para a reconfiguração do problema do *benchmark*, foram propostos três cenários diferentes [Heiming, 2006]:

- 1- Válvula V1 está fechada e bloqueada;
- 2- Válvula V1 está aberta mas bloqueada;
- 3- Válvula V1L está aberta, de modo a simular uma fuga no tanque 1;

A tarefa de reconfiguração é encontrar os parâmetros de um controlador, de modo a manter o nível de água no tanque 3 no valor do *SetPoint* indicado pelo utilizador, com uma perda de água mínima.

4.2 Simulador para modelo de três tanques (“*three-tank benchmark*”)

Um dos objectivos deste simulador era pô-lo a funcionar com o processo PCT-9, mas infelizmente o processo PCT-9 não estava a funcionar correctamente. Decidiu-se que, como iria ser encomendado um novo e mais moderno processo de tanques, seria melhor preparar o simulador para actuar com o novo processo. A sua entrega foi efectuada mais tarde do que o previsto e não foi possível utilizar o novo processo de tanques, pelo que se optou por outra alternativa. Foi decidido que seria feito um simulador, mas com uma comunicação, entre o *LabVIEW* e o *MATLAB*, TCP/IP, de modo a ser possível efectuar testes como se estivéssemos com um processo real.

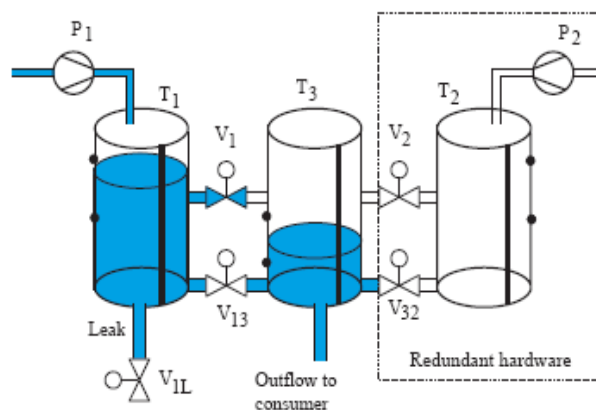


Figura 4.2 - Modelo de três tanques [Heiming, 2006].

Uma vez que o tempo que restava para a conclusão do trabalho se estava a esgotar, concluiu-se ser mais rápido implementar o simulador em *Simulink*, porque podiam ser programadas funções em *MATLAB* para depois comunicarem com o *Simulink* através das Funções “S”. Este modelo é ilustrado na Figura 4.3.

O modelo disponível sofreu algumas alterações de modo a poder ser adapta ao modelo dos três tanques e ao simulador em *LabVIEW*. Foram mantidos os valores dos parâmetros dos controladores PID bem como a estrutura “3TankSystem”. Foram adicionadas algumas válvulas, as Funções “S” e alguns elementos de visualização gráfica.

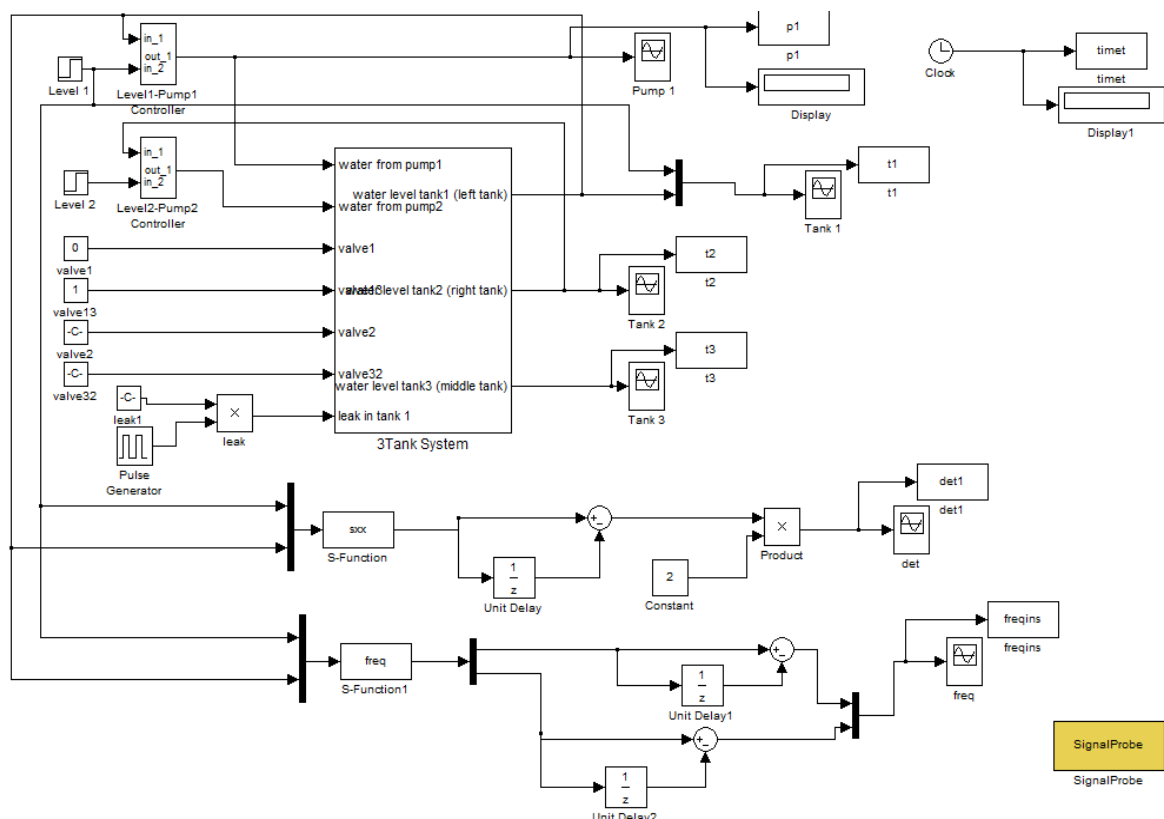


Figura 4.3 - Simulador em *Simulink*.

Foi criado um ficheiro .m que contém algumas variáveis globais. Estas variáveis ficam disponíveis durante toda a simulação. O uso de variáveis globais, neste caso, servirá para definir alguns parâmetros da simulação bem como alguns vectores. De seguida é apresentado o código do ficheiro que contém as variáveis globais.

```

% amain.m

clear all, close all, clc

global IAEvalue IAElimit kvzeros1 tvzeros1 e a det_osc kmax A d tx Ts Fs r N

kmax = 600; % duração da simulação

d=10; %tamanho da janela em amostras

A=1/100;

IAEvalue = zeros(kmax,1);

IAElimit = IAEvalue;

kvzeros1 = IAEvalue;

tvzeros1 = IAEvalue;

e = tvzeros1;

tx = e;

a = 1;

det_osc = IAEvalue;

Fs = 1; Ts = 1/Fs; r=32; N = length(e);

whos

```

Na Figura 4.4 é possível visualizar o interior do bloco *Level1 – Pump1 controller*. Dentro deste bloco existem as duas variáveis de entrada *in_1* e *in_2* que correspondem a *water level tank 1* (que é a água que se encontra no tanque 1) e *Level 1* (nível á agua inicial no tanque 1). Há um bloco *sum* para somar as duas variáveis de entrada, de modo a se poder alterar o nível de água no tanque 1. Existe um bloco *PID Controller* onde se encontram os parâmetros do controlador PID para controlar o nível de água no tanque 1. Estes parâmetros foram retirados do modelo já existente; existe ainda o bloco *Saturation* que é usado para limitar os valores; e o *out_1* que é o output do bloco.

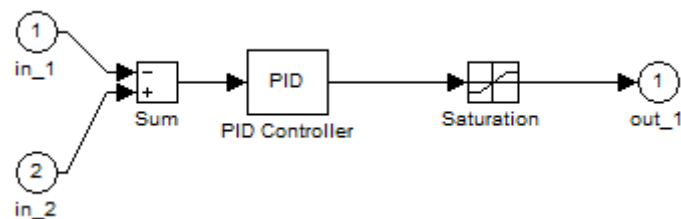


Figura 4.4 - Bloco *Level1 – Pump 1 controller*.

Na Figura 4.5 é possível visualizar o interior do bloco *Level2 – Pump2 controller*. Este bloco tem um funcionamento igual ao anterior, mas em vez de ser para o tanque 1 é para o tanque 2. Dentro deste bloco existem as duas variáveis de entrada *in_1* e *in_2* que correspondem a *water level tank 2* (que é a água que se encontra no tanque 1) e *Level 2* (nível de água inicial no tanque 2); há um bloco *sum* para somar as duas variáveis de entrada de modo a se poder alterar o nível de água no tanque 1. Existe um bloco *PID Controller* onde se encontram os parâmetros do controlador PID para controlar o nível de água no tanque 2; existe ainda o bloco *Saturation* que é usado para limitar os valores; e o *out_1* que é o output do bloco.

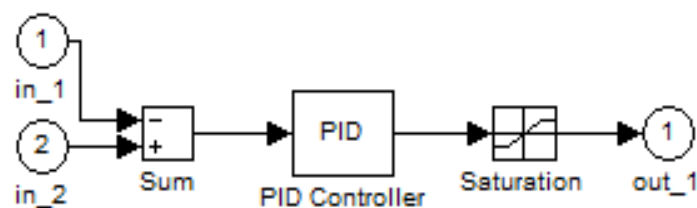


Figura 4.5 - Bloco *Level2 – Pump 2 controller*.

A Figura 4.6 elucida o interior do bloco *3Tank System*. Neste bloco existem sete variáveis de entrada, em que:

- Water from pump1, que como o próprio nome indica é o valor do caudal de água que sai da bomba 1;
- water from pump2, é o valor do caudal de água que sai da bomba 2;
- valve 1, é o valor da válvula 1 (se for 1 está aberta, se for 0 está fechada);
- valve 13; valve 2; valve 32; leak in tank1, que corresponde ao valor da válvula que simula uma fuga no Tanque 1.

Estas variáveis de entrada vão todas para o bloco *Mux* para de seguida entrarem numa *s-function mytank*, onde estão programadas todos os modelos dos tanques em *MATLAB*. No final são devolvidos os novos valores das alturas nos três tanques (h_1 , h_2 , h_3).

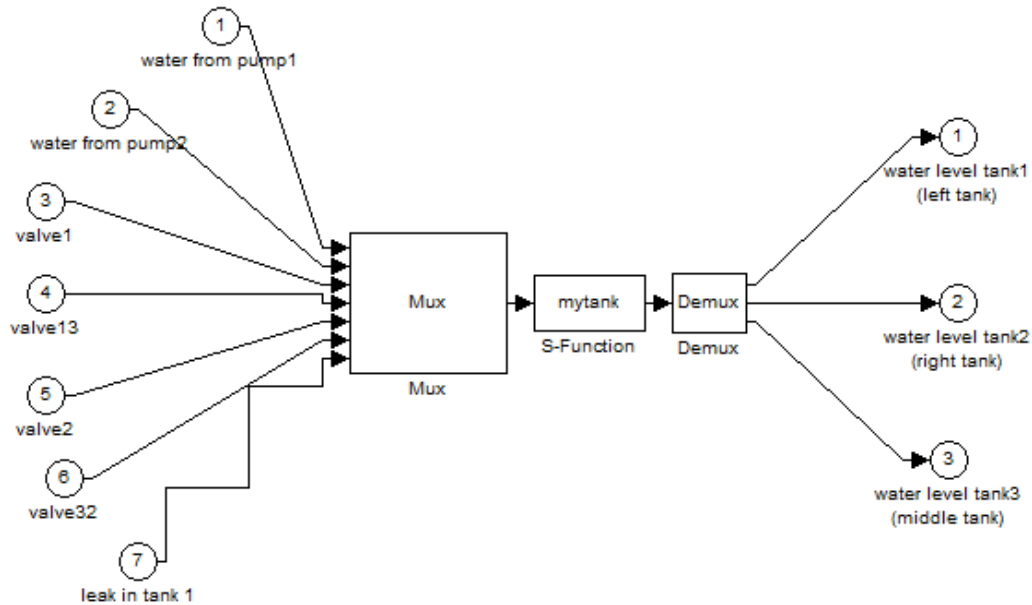


Figura 4.6 - Bloco 3Tank System.

Na Figura 4.7 apresenta-se um conjunto de blocos para a detecção de oscilações no Tanque 1.

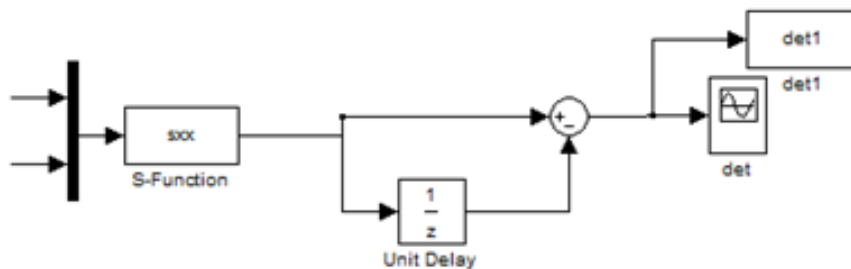


Figura 4.7 - Blocos para detecção de oscilações do tanque 1.

Para a detecção de oscilações foi usado o algoritmo do Integral do erro absoluto (IAE – *Integral of Absolute Error* [Kara, 2010] e [Ooi, 2008]). Esta detecção de oscilações funciona do seguinte modo: o utilizador coloca um certo valor de *SetPoint* para o Tanque 1e como o Tanque 3 tem de ter sempre água, o controlador faz com que o nível do Tanque 1 se mantenha no valor do *SetPoint*; sempre que o valor do nível de água passar pelo valor do *SetPoint*, há uma detecção de oscilação. Foi usada uma janela deslizante para facilitar os cálculos, visto que para calcular o integral do erro absoluto (IAE) e o limite do integral do erro absoluto (*IAElimit*) é necessário ter valores passados. De seguida é apresentado o código da função usada na detecção de oscilações.

```

function sys = Update(t,x,u)

global IAEvalue IAElimit kvzeros1 tvzeros1 e a det_osc A d Fs

if (t == 0),
    sys(1) =0;
    return;
end;

if ((t > 0)&& (t<=d))
    sp = u(1);
    y = u(2);
    e(t) = sp - y;
end

if (t > d)
    sp = u(1);
    y = u(2);
    e(t) = sp - y;
    if (((e(t-d) < 0) && (e(t-d+1) > 0)) || ((e(t-d) > 0) && (e(t-d+1) < 0)))
        if a>=2
            if (kvzeros1(a-1) == t)
                a=a-1;
            else
                kvzeros1(a) = t;
                tvzeros1(a) = t*0.1;
            end
        else
            kvzeros1(a) = t;
            tvzeros1(a) = t*0.1;
        end
    end
    if (a > 2)
        for i=kvzeros1(a-2):kvzeros1(a-1)
            isoma = abs(mean([e(i-1) e(i)]))*0.1;
            IAEvalue(a) = IAEvalue(a-1) + isoma;

            Tu = abs(tvzeros1(a) - tvzeros1(a-2));
            wu = (2*pi)/Tu;
            IAElimit(a) = (2*A)/wu;
        end
    end
end

```

```

end

if (IAEvalue(a) >= IAElimit(a))

    det_osc(t) = 1;

else

    det_osc(t) = 0;

end

end

a = a + 1;

end

sys(1) = det_osc(t);

else

    sys(1) = 0;

end

end

```

Existem duas variáveis de entrada: *Level 1*, que corresponde ao valor do *SetPoint* do Tanque 1 e *water level tank1*, que corresponde ao nível de água no tanque 1; existe ainda uma Função “S” onde se encontra todo o código em *MATLAB* para a detecção de oscilações. À saída da Função “S” temos um *Unit Delay* que contém o valor anterior, que é subtraído ao actual de modo a se obter o valor desejado, isto porque as Função “S” devolvem estados que vão ser sempre incrementados ao estado anterior, daí ser necessário efectuar subtracção.

A Figura 4.8 apresenta um conjunto de blocos que nos permite obter a frequência instantânea do erro de controlo do tanque 1.

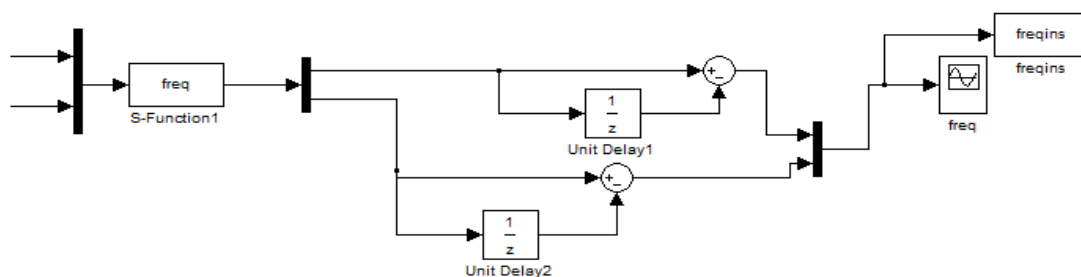


Figura 4.8 - Blocos para o cálculo da frequência instantânea.

Para o cálculo da frequência instantânea do erro, foi usada uma função do *MATLAB*, a função *hilbert* (para os cálculos da transformada de Hilbert). Esta função recebe os valores do *SetPoint* e o valor do nível de água do tanque1, calcula o erro entre estes dois valores e aplica a transformada de hilbert ao valor do erro. Após obtermos o valor da transformada de hilbert do erro, é calculada a frequência instantânea em Hz.

O código desta função é apresentado de seguida.

```
function sys = Update1(t,x,u)
global e Fs r N
if (t == 0),
    sys(1) = 0;
    sys(2) = 0;
    return;
end;
if ((t > 0) && (t<=r))
    sp = u(1);
    y = u(2);
    e(t) = sp - y;
end
if (t>32)
    sp = u(1);
    y = u(2);
    e(t) = sp - y;
    z = hilbert(e);
    f = angle(z(2:N).*conj(z(1:N-1))) / (2*pi); % mult by Fs to get Hertz
    fi = Fs * f; % instantaneous frequency [Hz]
    sys(1) = abs(fi(t-r));
    sys(2) = e(t-r);
else
    sys(1) = 0;
    sys(2) = 0;
end
end
```


A Função “S” retorna dois estados: o da frequência instantânea e o do erro; para se obterem os valores desejados e tal como foi referido, é necessário, subtrair os valores anteriores de modo a se obter o valor da frequência instantânea ou do erro e não os seus estados.

De seguida será apresentado e descrito o simulador em *LabVIEW*.

O simulador em *LabVIEW* está dividido em duas partes, no painel frontal e no diagrama de blocos. A parte de programação é toda feita no diagrama de blocos, deixando assim a parte de visualização a cargo do painel frontal. Em primeiro lugar, no painel frontal foram colocados os tanques, as válvulas, as bombas de água, os gráficos de visualização, os parâmetros dos controladores e alguns alarmes. De seguida estes itens foram programados no diagrama de blocos, de modo a tornar possível ter o ambiente de visualização como desejado.

Na Figura 4.9 está ilustrado o Painel frontal do simulador e na Figura 4.10 está ilustrado o diagrama de blocos do simulador.

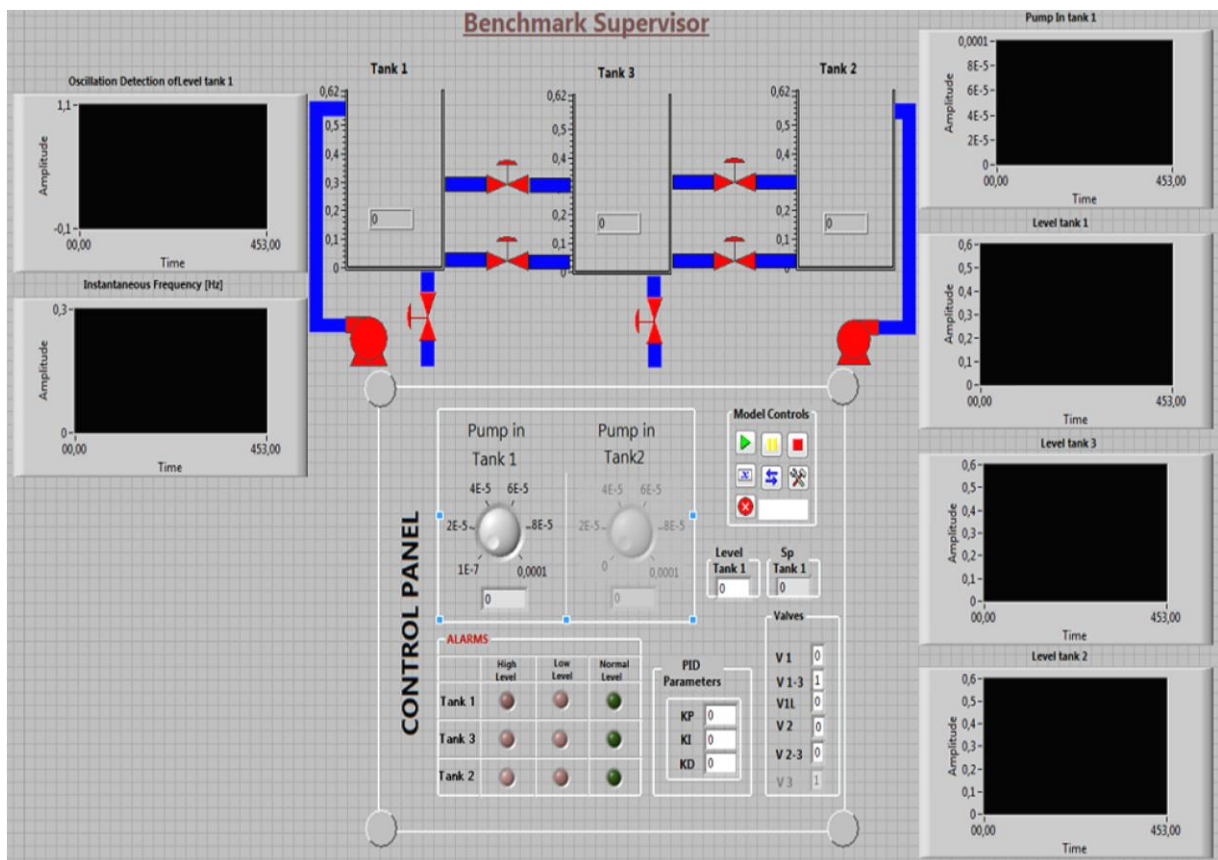


Figura 4.9 - Painel Frontal do simulador em *LabVIEW*.

gráfico que mostra a frequência instantânea associada à variação da altura do tanque 1 e por fim, o gráfico da variação da altura do Tanque 2).

Verifica-se também que é possível o utilizador abrir ou fechar as válvulas, mudar os valores dos parâmetros do controlador PID e visualizar os alarmes que mostram se os tanques estão muito cheios, muito vazios ou se estão com um nível de água nominal.

4.3 RESULTADOS EXPERIMENTAIS

Neste subcapítulo apresentam-se os resultados experimentais para o modelo dos três tanques (“three-tank benchmark”, Figura 4.1).

Este subcapítulo encontra-se dividido em duas secções. Na primeira secção é feita a descrição da simulação, com a indicação dos vários parâmetros utilizados. Na segunda, são apresentados os resultados obtidos na simulação.

4.3.1 Descrição da Simulação

Para simular o modelo dos três tanques (“three-tank benchmark”, Figura 4.2), foram usados dois computadores, um deles com o *MATLAB/Simulink* aberto, onde está a correr o programa principal, e outro com o *LabVIEW*. Os dois computadores estão ligados à *Internet* e comunicam-se entre si através do SIT.

Foram utilizados os seguintes parâmetros de simulação:

- Tempo de simulação = 600s;
- Nível de água inicial no tanque 1, $h_1(t_0) = 0,3$;
- *SetPoint* = 0,3;
- Nível de água inicial no tanque 3, $h_3(t_0) = 0$;
- Nível de água inicial no tanque 2, $h_2(t_0) = 0,2$;
- Parâmetros do controlador PID:
 - $K_p = 0,001$;
 - $K_I = 6,135 \times 10^{-6}$;
 - $K_D = 0,00001$;

- Válvulas:
 - $V1=1$ (válvula aberta);
 - $V3=0$ (válvula fechada);
 - $V1L=1$ (Esta é controlada por um relé que abre e fecha de modo a simular uma falha inconstante);
- A Bomba P1 está ligada e a água só entra pelo tanque 1.

4.3.2 Resultados Obtidos

Depois de efectuada a simulação, os resultados obtidos foram bastante satisfatórios. Em *LabVIEW* foi possível visualizar e controlar como se estivéssemos em *Simulink*, isto é, conseguiu-se monitorizar toda a simulação através do *LabVIEW*. Nas Figuras seguintes (4.11 e 4.12) é possível observar e comparar o que foi visualizado nos dois computadores.

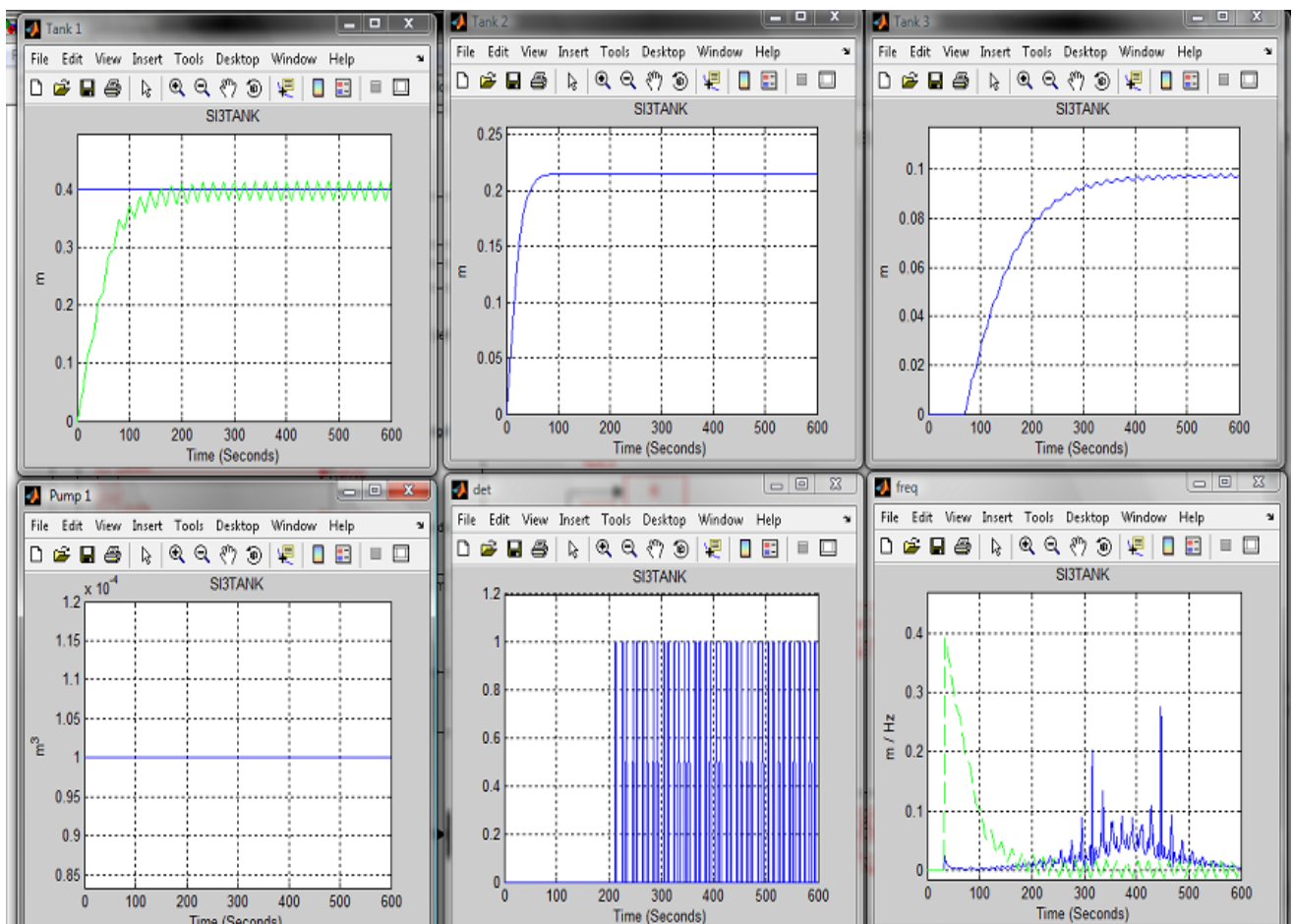


Figura 4.11 - Gráficos em *MATLAB/Simulink* dos resultados da simulação.

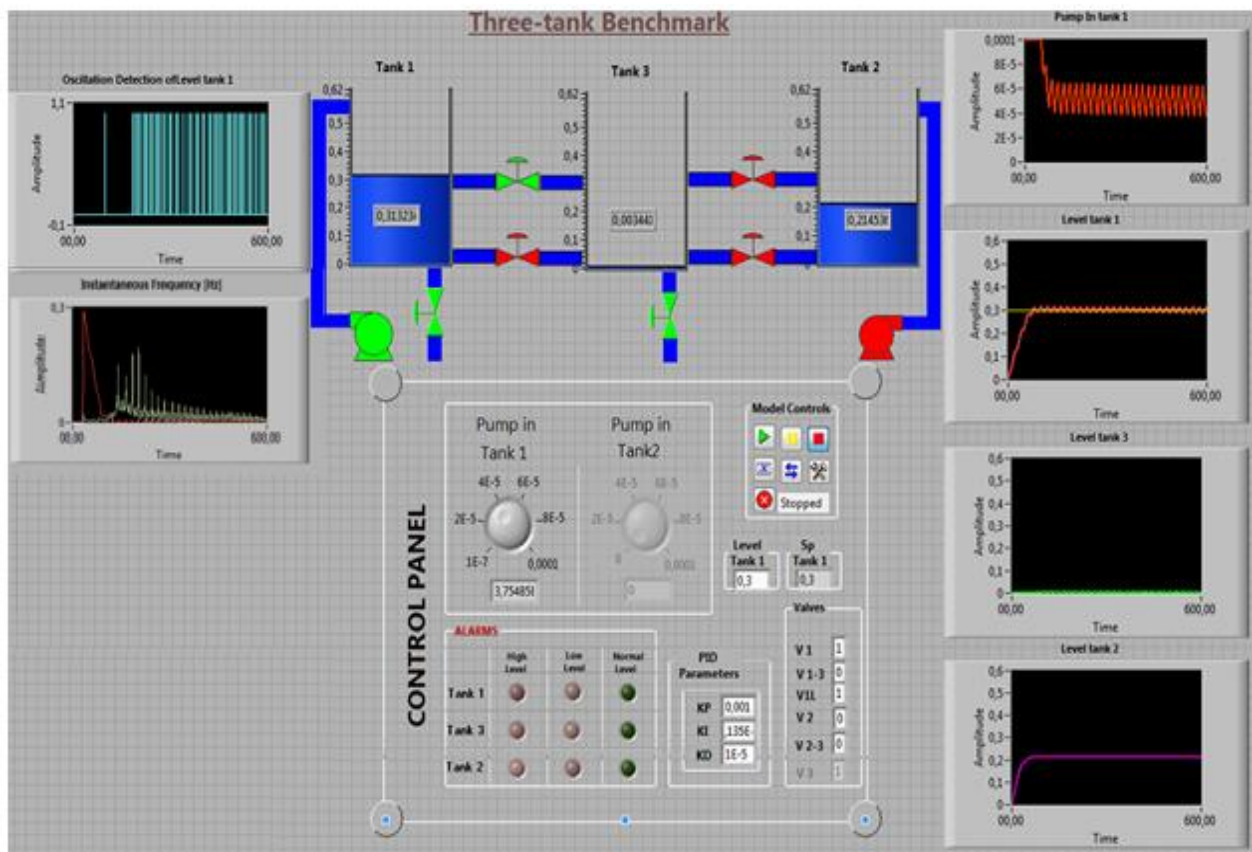
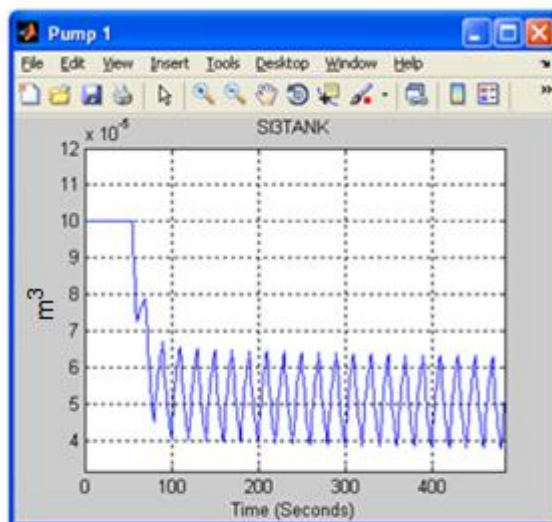
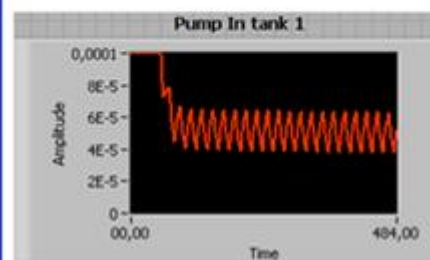


Figura 4.12 - Gráficos em *LabVIEW*.



a)



b)

Figura 4.13 - Gráfico da bomba 1 em: a) *Simulink*; b) *LabVIEW*.

Analisando com mais pormenor verifica-se que na Figura 4.13 podemos acompanhar o desempenho da bomba 1. Como pode ser observado as oscilações provocadas pela válvula V11, começam a se sentir aos 90s que é o tempo que leva a água no tanque 1 a atingir os 0,30 m (valor do *SetPoint*). Este facto pode ser confirmado na Figura 4.14 que mostra o enchimento do tanque 1.

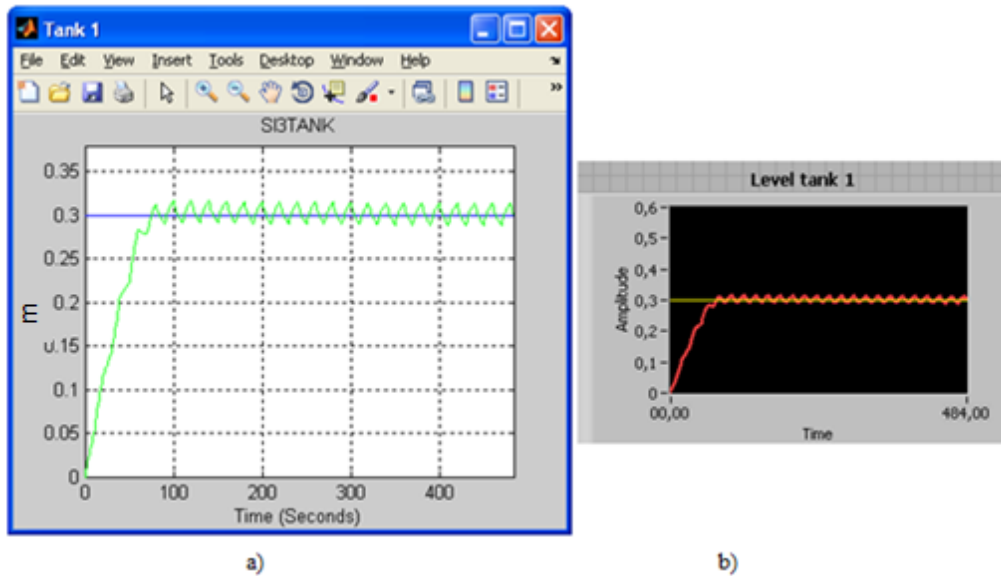


Figura 4.14 - Gráfico do nível do tanque 1 em: a) *Simulink*; b) *LabVIEW*.

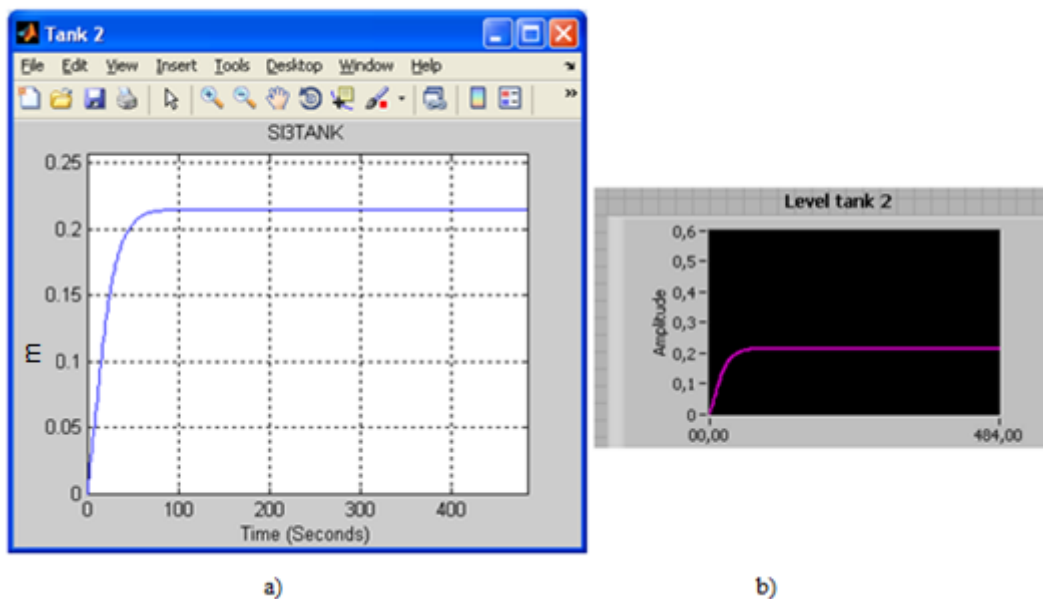


Figura 4.15 - Gráfico do nível do tanque 2 em: a) *Simulink*; b) *LabVIEW*.

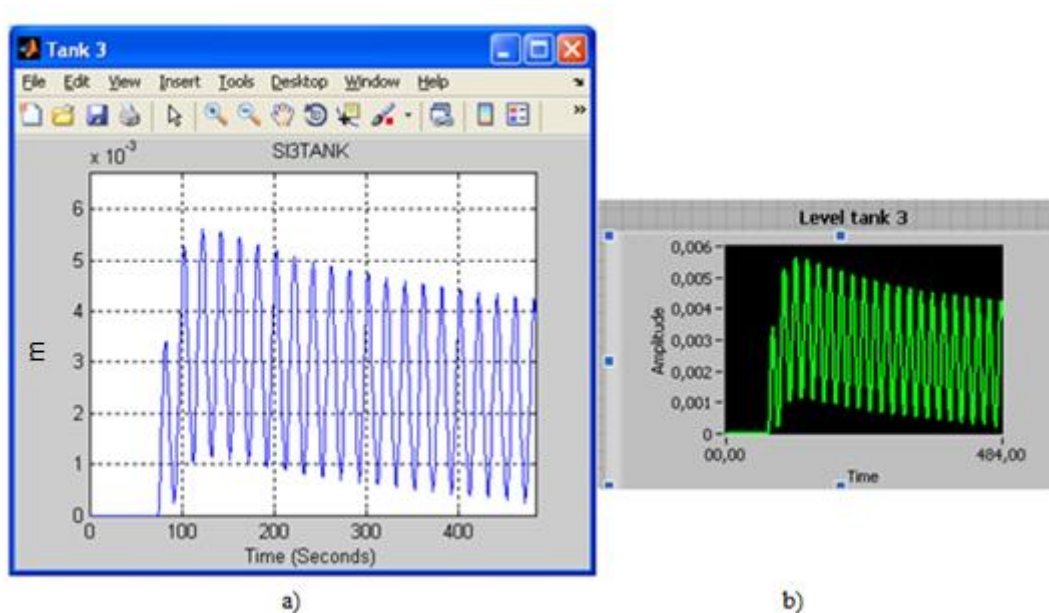


Figura 4.16 - Gráfico do nível de água no tanque 3 em: a) *Simulink*; b) *LabVIEW*.

Na Figura 4.16 pode ser acompanhado o enchimento do tanque 3. Só a partir dos 90s é que começa a entrar água para o tanque 3, pois é nesta altura que o nível de água do tanque 1 chega aos 30cm que é a altura a que se encontra a válvula V1.

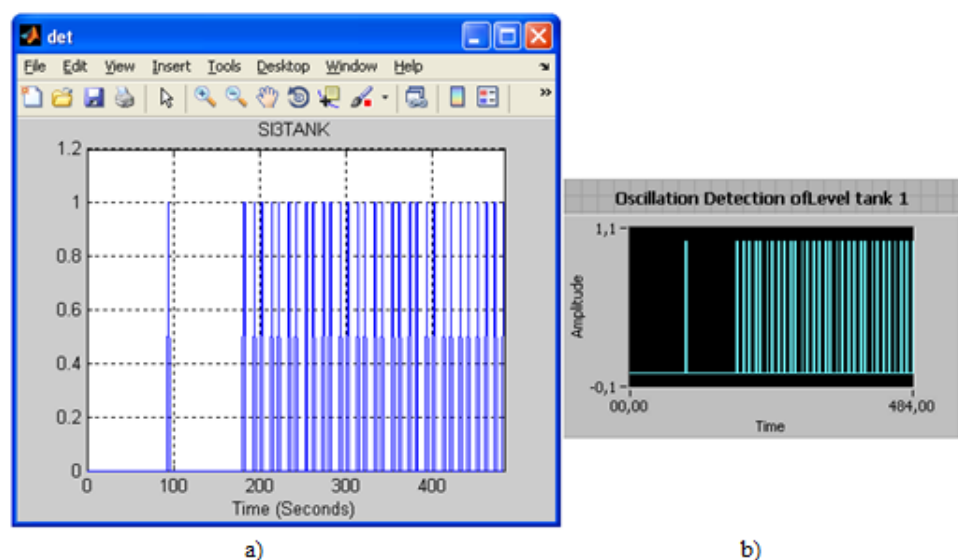


Figura 4.17 - Gráfico da detecção de passagens pelo "SetPoint" em: a) *Simulink*; b) *LabVIEW*.

Na Figura 4.17 pode ser observada a detecção de oscilações no tanque 1. Como foi usado uma janela deslizante de tamanho 10, só 10s depois de haver a primeira oscilação é que essa oscilação é detectada, por exemplo, a primeira vez que há uma detecção de

oscilação é quando o nível de água do tanque 1 atinge os 30cm e isso acontece aproximadamente aos 90s por isso é que só aos 100s é que ocorre a primeira detecção na Figura 4.17.

As próximas oscilações só são detectadas perto dos 200 s devido ao limite do Integral do erro absoluto ter um valor mais elevado do que o valor máximo de uma oscilação, logo como a onda sinusoidal nunca passa o valor do limite, a detecção não é detectada.

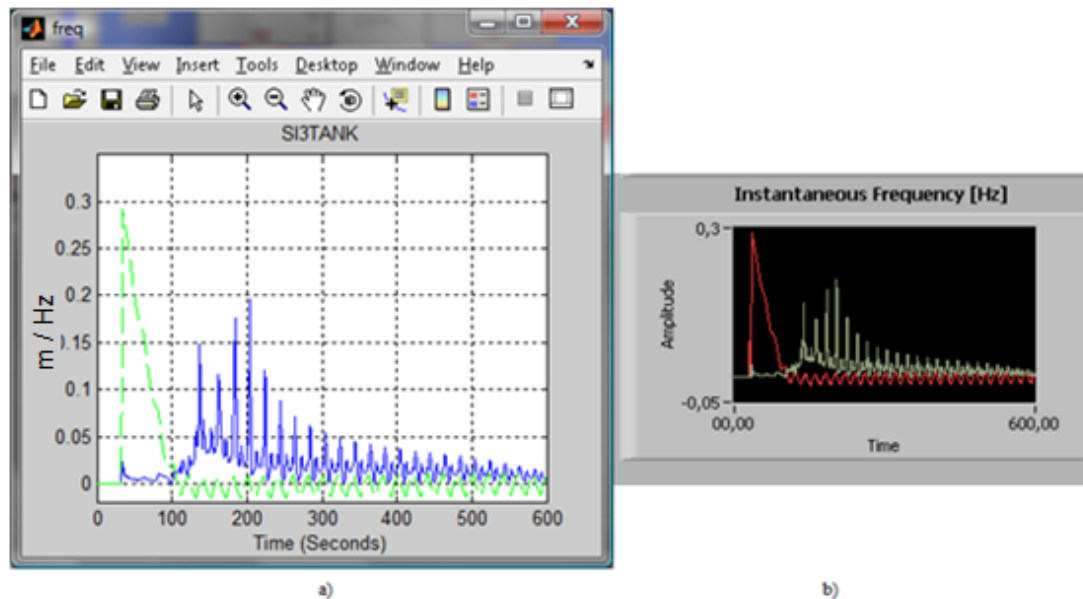


Figura 4.18 - Gráfico da frequência instantânea em: a) *Simulink* (a azul a frequência instantânea e a verde o erro); b) *LabVIEW*.

Na Figura 4.18 é ilustrada a frequência instantânea do erro. Para uma melhor visualização, foi usado um filtro passa-baixo para filtrar o sinal do erro, de modo a ter um sinal de frequência sem os picos. Mesmo assim são ainda visíveis alguns picos referentes aos valores iniciais do erro de controlo.

Da análise das Figuras anteriores, verifica-se que os gráficos em *Simulink* são idênticos aos gráficos em *LabVIEW*, que é possível observar em *LabVIEW* o mesmo que se observa em *Simulink*, e que é possível também controlar toda a simulação em *LabVIEW*.

5 CONCLUSÕES

Neste último capítulo são apresentadas as considerações finais sobre o trabalho realizado, as principais contribuições desta dissertação, e são ainda identificadas possíveis áreas de trabalho futuro.

5.1 Considerações Finais

Nesta dissertação foi abordado um tema que nos últimos anos tem sido alvo de um crescente interesse no seio da comunidade científica: os laboratórios remotos. Foi proposta uma arquitectura de laboratório remoto, na qual era composta por um bloco Supervisor, por um bloco Rede – TCP/IP e por um bloco Anel de Controlo como mostrado na Figura 3.1. Decidiu-se que iria ser usado o *MATLAB/Simulink* para programar todo o anel de controlo e o *LabVIEW* para a parte do supervisor.

O bloco Supervisor analisa o desempenho da instalação (detecta oscilações do anel de controlo), analisa o desempenho dos controladores e também faz a monitorização da instalação, ou seja, o bloco Supervisor tem acesso aos valores dos parâmetros dos controladores, aos valores da referência do anel de controlo e aos valores de entrada e de saída da instalação. Através deste bloco é também possível o utilizador alterar alguns parâmetros referentes à instalação. Todo este bloco encontra-se programado em *LabVIEW*, por ser um programa que permite ao utilizador ter uma boa visualização gráfica.

O bloco Rede – TCP/IP é o bloco responsável por toda a comunicação entre o bloco Supervisor e o bloco Anel de controlo. Toda a comunicação entre o bloco Supervisor e o bloco Anel de Controlo é feita através do *SIT Connection Manager*.

O bloco Anel de Controlo contém toda a instalação, bem como os controladores que controlam essa mesma instalação. Todo este bloco encontra-se programado em *MATLAB/Simulink*.

Surgiram algumas dificuldades durante a realização do trabalho desta dissertação, mas o resultado final foi bastante satisfatório, ou seja, foram atingidos os objectivos. O simulador/controlador foi criado com sucesso, permitindo ao utilizador estudar as características do modelo dos 3 tanques e analisar as dinâmicas do sistema, como se estivesse a trabalhar com o processo real, deparando-se com eventuais falhas que possam acontecer durante as simulações.

O simulador/controlador apresentado nesta dissertação poderá vir a ser utilizado por estudantes, professores e engenheiros, constituindo uma mais-valia para o desempenho das suas actividades.

Embora seja necessário efectuar alguns ajustes no supervisor apresentado e uma vez que é possível, utilizando o *Simulink* e o *LabVIEW*, criar um supervisor para outros processos ou situações, conclui-se que esta dissertação contribuiu para o progresso dos laboratórios remotos.

5.2 Trabalho Futuro

As modificações que são necessárias efectuar para que o processo real possa comunicar com o programa em *Simulink*, deviam ser objecto de estudo e de trabalhos futuros.

Enumeram-se em seguida algumas sugestões que se considera poderem vir a enriquecer o trabalho realizado.

- Para cenários industriais, este simulador necessita da ligação entre o programa em *MATLAB/Simulink* e a linha de montagem. Uma vez feita esta ligação, e adaptada uma VI a esse programa em *Simulink* teremos um acesso remoto à linha de montagem.
- Para cenários de educação/formação, é necessária a ligação do simulador em *MATLAB/Simulink* ao processo em si, de modo a que este possa ser controlado remotamente através do *LabVIEW*.

Na Figura 5.1 está ilustrada uma possível arquitectura para um trabalho futuro.

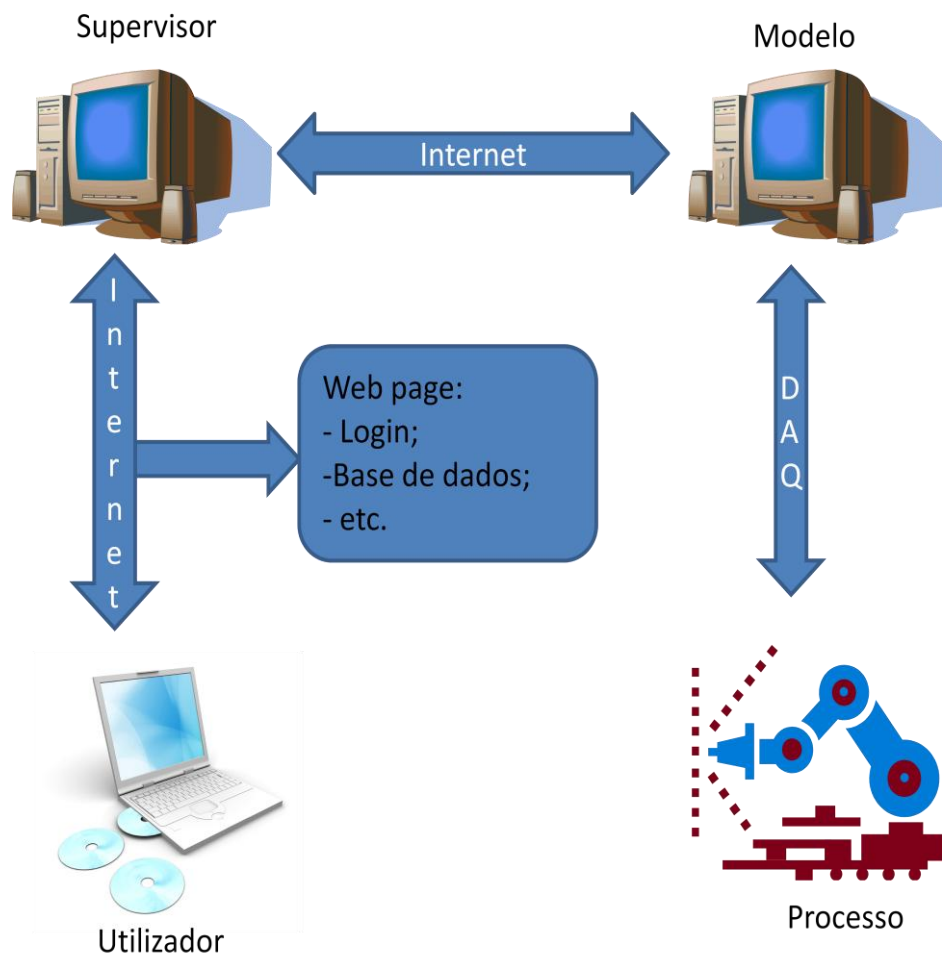


Figura 5.1 - Arquitectura proposta para trabalho futuro.

Estes desenvolvimentos seriam muito úteis, quer ao nível do ensino à distância, quer a nível industrial, proporcionando neste último caso, uma maior facilidade no controlo/monitorização de processos industriais, que poderia ser efectuada à distância, com todas as vantagens.

Apêndice A – Simulador de processo

PCT-9

Neste apêndice é apresentado um laboratório remoto para um processo laboratorial. Este laboratório remoto era a ideia inicial desta dissertação. Numa primeira abordagem, foi sugerido para a simulação o processo PCT-9 (Armfield) baseado no modelo *benchmark* dos três tanques. Na Figura A.1 é possível observar a arquitectura do processo, bem como os seus componentes.

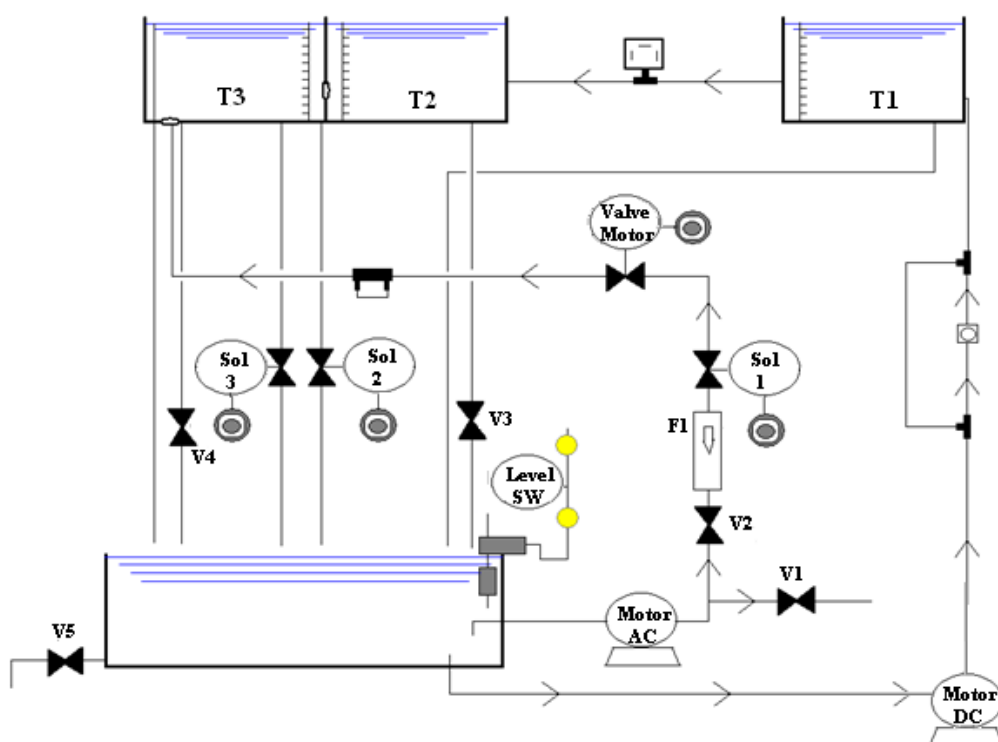


Figura A.1 - Esquema do processo PCT-9.



Figura A.2 - Processo PCT-9.

Para poder ser utilizado o equipamento PCT-9 correctamente foi necessário efectuar a substituição das tubagens, bem como a implementação de alguma electrónica para um sensor de nível do tipo potenciómetro e para um sensor de caudal com o objectivo de calcular o fluxo de água que passa nos tubos de ligação de um tanque para o outro e á saída da bomba.

Para a electrónica do sensor de nível do tipo potenciómetro foi usado um regulador 7809 para regular a tensão de entrada do circuito, isto é, passando-a de 12V para 9V, um AMPOP - LM 358N e uma resistência de 10k Ω .

A Figura A.3 ilustra a electrónica implementada e na tabela 4 podemos verificar os testes efectuados.

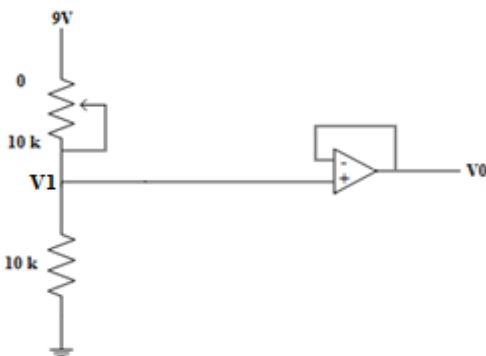


Figura A.3 - Electrónica do sensor de nível do tipo potenciómetro.

À medida que o sensor vai subindo ou descendo, o valor do potenciômetro vai-se alterando (0-10 k), originando um diferente valor de V_0 (4,5 - 9V), cada valor de V_0 , corresponde a uma determinada altura.

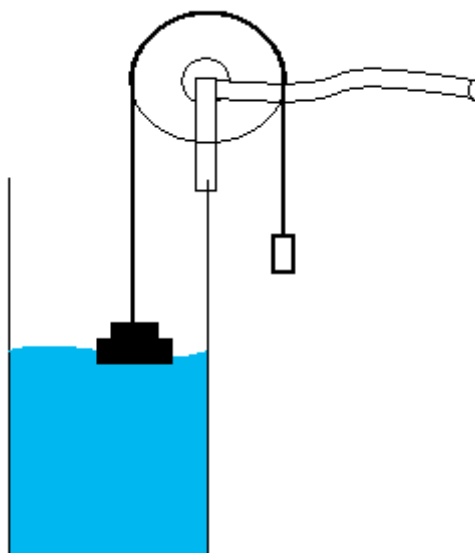


Figura A.4 - Sensor de nível com bóia.

Tabela A.1 - Valores dos testes efectuados.

| Nível | Potenciômetro | V_1 (V) | V_0 (V) |
|--------|---------------|-----------|-----------|
| Máximo | 0 | 9 | 9 |
| Mínimo | 10 k | 4,5 | 4,5 |
| Médio | 5 k | 6 | 6 |

O sensor de caudal (Farnell 178-523) necessitava de um conversor frequência/tensão para medir o fluxo de água que passa pelos tubos de água. O sensor, envia um determinado valor em frequência, em função da quantidade de água que está a passar. O conversor converte essa frequência em tensão, de modo a tornar mais fácil o uso desse valor nos cálculos.



Figura A.5 - Sensor (Farnell 178-523) de fluxo de água.

A electrónica do conversor que foi utilizada pode ser visualizada na Figura A.6.

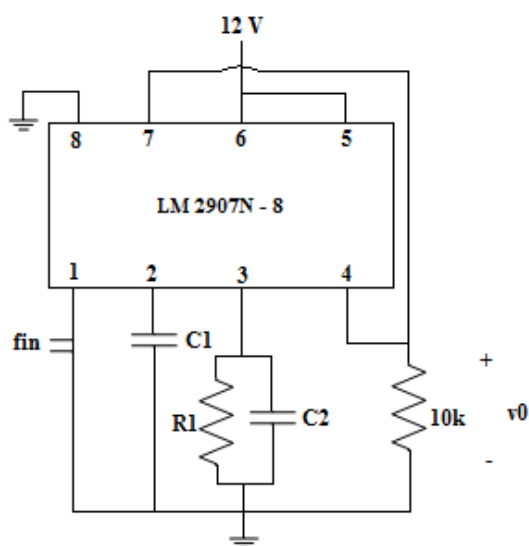


Figura A.6 - Electrónica do conversor F/V.

Foi criado um simulador em *LabVIEW* com o propósito de se ter um simulador a funcionar de acordo com o processo dos tanques PCT-9 modificado para 3 Tanques.

Este simulador pode ser visualizado nas Figuras A.7 e A.8.

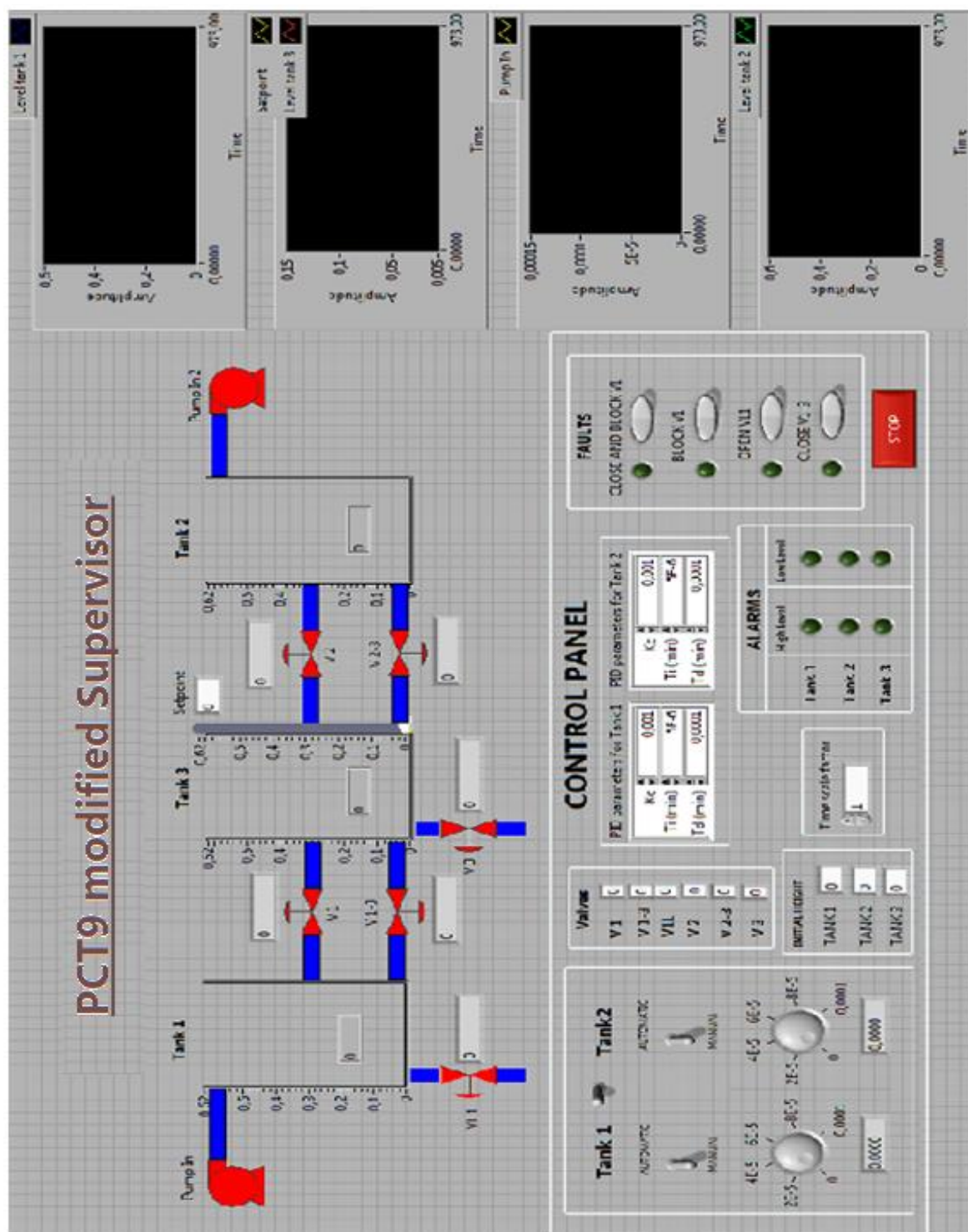


Figura A.7 - Painel frontal do simulador do PCT-9 modificado.

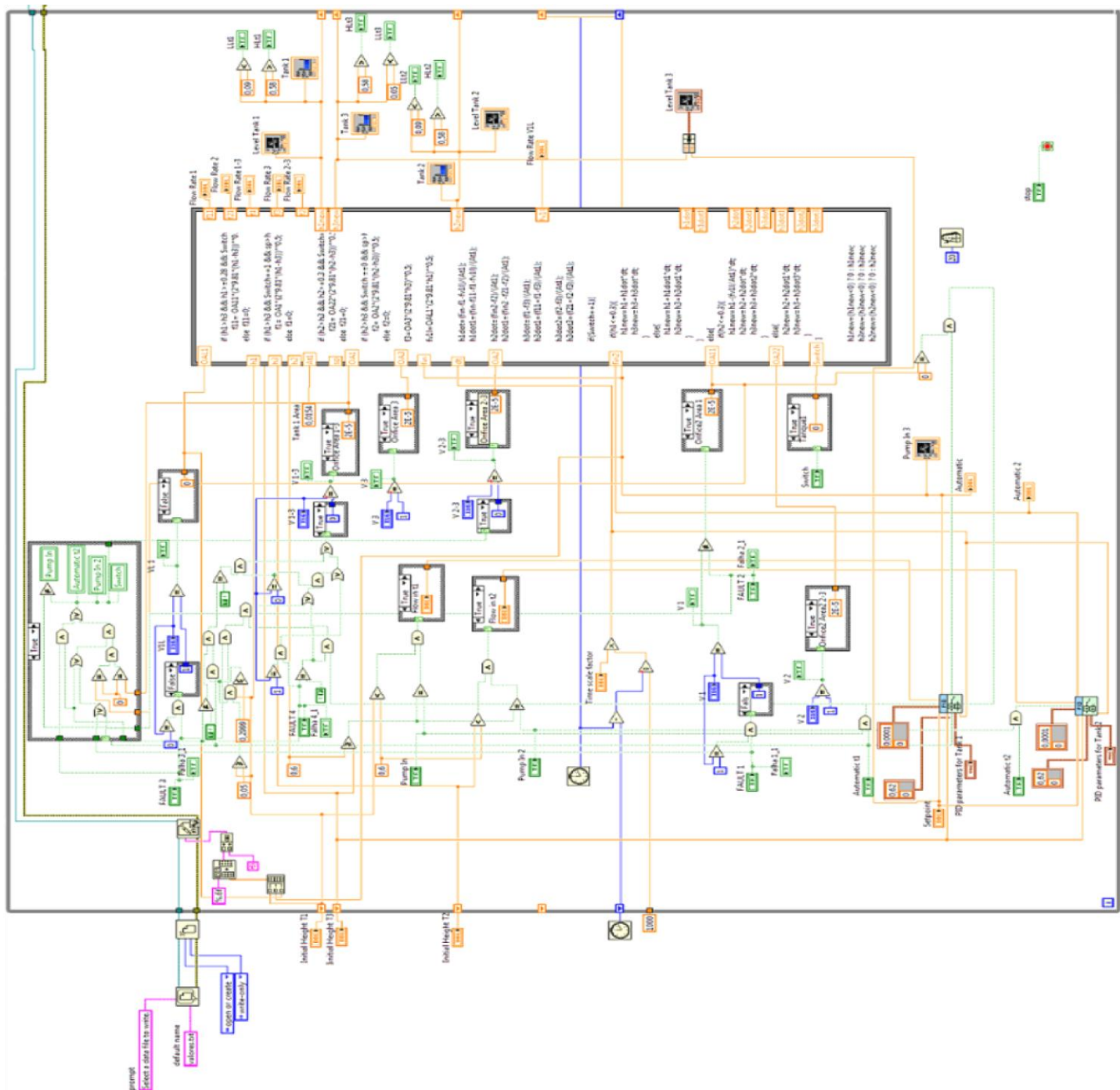


Figura A.8 - Diagrama de blocos do simulador do PCT-9 modificado.

A Figura A.8 é apenas apresentada parte do código de implementação do simulador para que o mesmo se apresente legível.

Neste simulador o enchimento do tanque 3 (central) é feito por um tanque só, isto é, quando a bomba do tanque 1 está em funcionamento, a bomba do tanque 2 está

desligada e vice-versa. Para cada um destes tanques (1 ou 2) havia duas formas de utilização: automático e manual. Caso fosse escolhido o modo automático, o simulador efectuava o controlo sem ajuda do utilizador. Existia um *SetPoint* no tanque 3 e o controlo da quantidade de água que entrava nos tanques que abastecem o tanque 3, era feito pelo simulador, através de controladores PID. Os parâmetros dos controladores foram retirados de um modelo feito em *Simulink*. Durante o processo de enchimento dos tanques, poderiam ocorrer falhas. As falhas eram devidas essencialmente ao mau funcionamento das válvulas:

- Falha 1 – Fechar e bloquear a válvula v1;
- Falha 2 – Bloquear a válvula v1;
- Falha 3 – Abrir a válvula v1l (válvula de fuga do tanque 1);
- Falha 4 – Fechar a válvula v1-3;

No caso de ocorrer alguma destas falhas, o simulador verificaria qual a melhor opção para manter o nível de água no tanque 3 igual ao valor do *SetPoint* que tinha sido escolhido, caso fosse necessário, o simulador mudaria o tanque de abastecimento (troca do tanque 1 para o 2).

No caso de ter sido escolhido o modo manual, o utilizador teria o controlo dos tanques: por exemplo do caudal de água que entrava nos tanques 1 e 2.

Com este simulador, como se pode observar pela Figura A.7, é possível também visualizar quatro gráficos: no primeiro, pode-se acompanhar o enchimento do primeiro tanque; no segundo, é possível ver o valor do *SetPoint*, e acompanhar o enchimento do tanque 3; no terceiro, acompanha-se o fluxo de água da bomba 1 e o quarto gráfico mostra o enchimento do tanque 2.

Apêndice B – Simulador de processo de três tanques da Amira

O novo processo dos tanques está ilustrado na Figura B.1 e ao simulador anterior (para o processo PCT - 9) foram feitas algumas alterações de modo a ter um simulador igual ao processo.



Figura B.1 - Novo processo DTS-200 Amira.

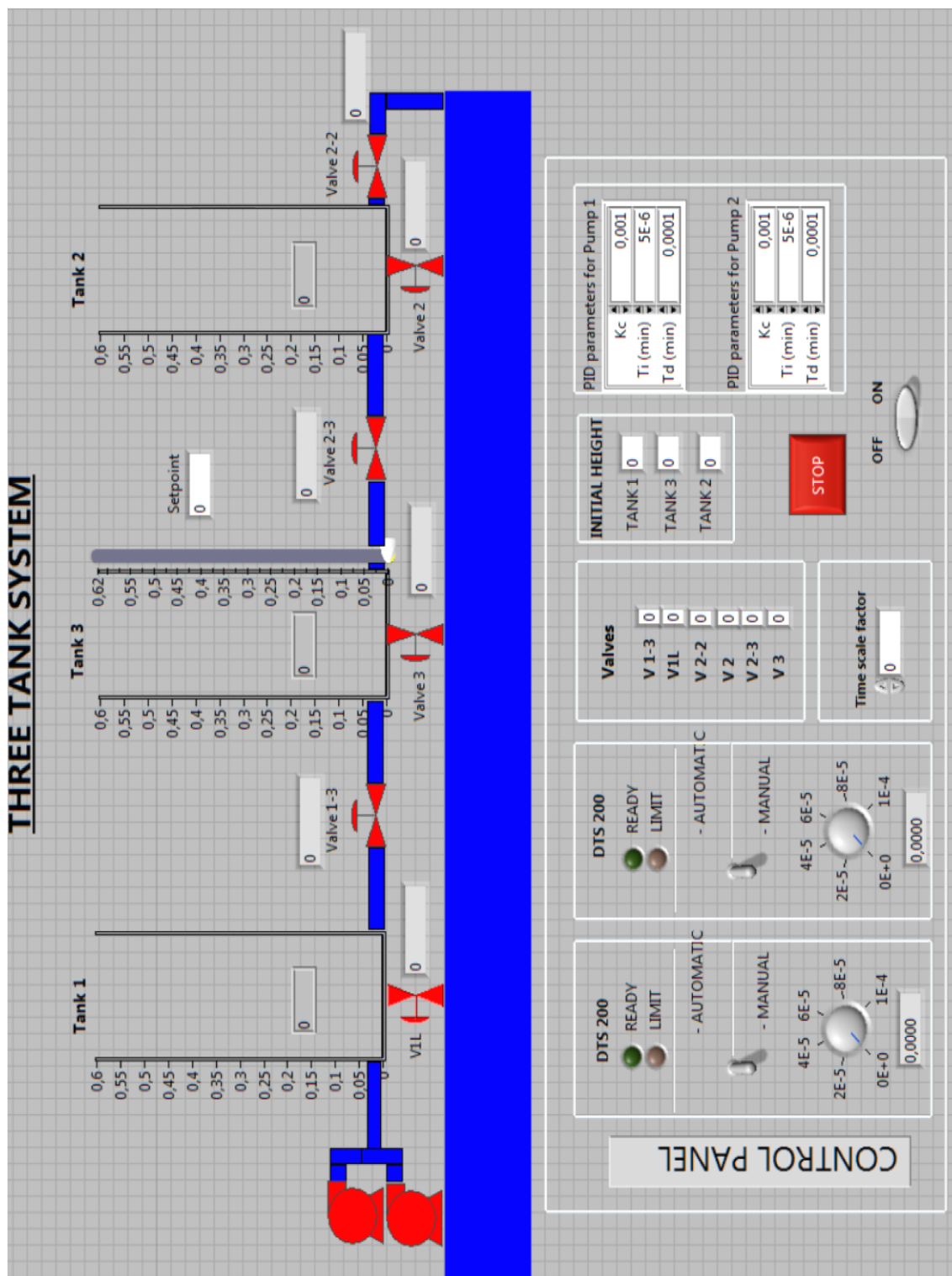


Figura B.2 - Novo painel frontal para o novo processo de tanques.

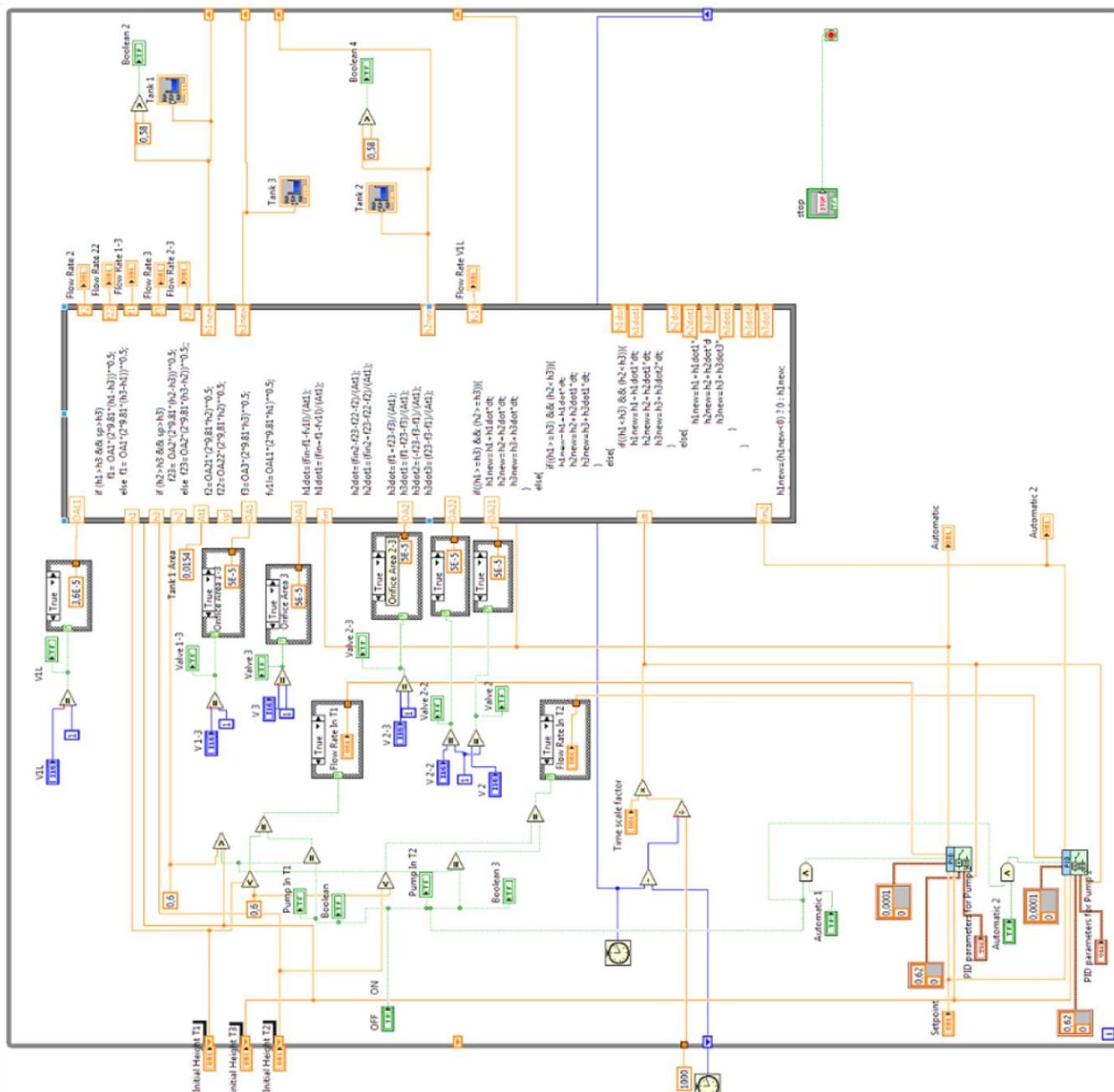


Figura B.3 - Novo diagrama de blocos para o novo processo de tanques.

Para este novo simulador, foi necessário alterar o número de válvulas entre os tanques, as áreas dos tanques e o diâmetro dessas mesmas válvulas. Em relação às equações utilizadas para o enchimento dos tanques, são as mesmas que as usadas no simulador anterior (para o processo PCT - 9).

REFERÊNCIAS BIBLIOGRÁFICAS

[André, 2004] M. André, C. Marco and F. Elnatan, *Laboratório Remoto aplicado ao Ensino de Engenharia Eletrônica*, Campinas, Brasil, 2004.

[Aktan, 1996] Aktan, B., *Distance learning applied to control engineering laboratories*. In IEEE Transaction on Education, New York, 1996.

[Apkarian, 2000] Apkarian, J., Dawes, A., *Interactive control education with virtual presence on the web*. In American Control Conference, Chicago, 2000.

[Asleson, 2006] Asleson, R., Schutta, N. *Foundations of Ajax*. Apress, 2006.

[Astrom, 1997] Åstrom, K., Wittenmark, B., *Computer Controlled Systems: Theory and Design*, Prentice-Hall, 1997.

[Bing, 2004] Bing, D., Keck-Voon, L., Habib Mir, M., *Developing and implementing online laboratory for control engineering education*. In International Conference on Control, Automation, Robotics & Vision, China, 2004.

[Bonavita, 2006] Bonavita, N., Caserza, J. e Martini, R., *Control Performance Monitoring: A Data-Driven Approach*, Milan, 2006.

[Buhler, 2000] Buhler, D.. The virtual automation lab-web based teaching of automation engineering concepts. In IEEE International Conference and Workshop on the engineering of Computer based systems, New York, 2000.

[**Canfora, 2004**] Canfora, G., Daponteb, P., Rapuano, S., *Remotely accessible laboratory for electronic measurement teaching*, Computer Standards & Interfaces, Vol. 26, pp. 489-499, 2004.

[**Casini, 2004**] Casini, M., Prattichizzo, D. e Vicino, A. *E-learning by remote laboratories: a new tool for control education*. In: IFAC Symposium on Advances in Control Education, University of Oulu, Finland, 2004.

[**Chan, 2002**] Chan, C., Kwanr, R., Chan, J., *Learning Control Systems on the Web*. In: International Conference on Computers in Education (ICCE 02), 2002.

[**Cmuk, 2008**] Cmuk, D., Mutapcic, T., Borsic, M., *Mobile Measurement Support for Remote Laboratories and E-Learning Systems*. In: 16th IMEKO TC4 Symposium Exploring New Frontiers of Instrumentation and Methods for Electrical and Electronic Measurements, Florence-Italy, 2008.

[**Dormido, 2002**] S. Dormido, *Control learning: Present and future*. In: IFAC World Congr. Automatic Control, Barcelona, Spain, pp. 81-103, 2002

[**Eduardo, 2007**] Eduardo, M. *Laboratório remoto para ensino a distância de sistemas de controlo distribuído*, Dissertação de mestrado, Escola de Engenharia de São Carlos, Brasil, 2007.

[**Exel, 2000**] Exel, M. *Simulation workshop and remote laboratory: two web-based training approaches for control*. In: American Control Conference, Chicago, 2000.

[**García-Zubía, 2007**] García-Zubía, J., P. Orduña, D. López-de-Ipiña, U. Hernández, I. Trueba, *Remote Laboratories from the Software Engineering Point of View*, in Advances on remote laboratories and e-learning experiences. Univ. Deusto, Bilbao, 2007.

[**Heiming, 2000**] Heiming, B., Lunze, J., *Definition of the three-tank benchmark problem for controller reconfiguration*. P. M. Frank, 2000.

[Hua, 2003] Hua, J. e Ganz, A. *A new model for remote laboratory education based on next generation interactive technologies*, In Proceedings of ASEE New England Regional Conference, Orono, ME, May, 2003.

[Imbrie, 2005] Imbrie, K., Raghavan, S., *Work in progress – a remote e-laboratory for student investigation, manipulation and learning*. In: ASEE/IEEE Frontiers in Education Conference, New York, 2005.

[Isermann, 2006] Isermann, R. *Fault-Diagnosis Systems: An Introductcion from Detection to Fault Tolerance*, Springer, 2006.

[Karra, 2010] S. Karra, M. Jelali, M. Nazmul Karim and A. Horch, *Detection of Oscillations Control Loops*, Advances in Industrial Control, Part I, AIC, 2010.

[Ko, 2001] Ko, C., *Development of web-based laboratory for control experiments on a coupled tank apparatus*. In: IEEE Transaction on Education, New York, 2001.

[Laboucine, 1999] Laboucine, E., *Control Loop Performance Assessment and Oscillation Detection*, in Electrical and Computer Engineering, UBC, 1999.

[Leitão, 2006] Leitão, S. *Um Ambiente Experimental à Distância*, Tese de mestrado. Faculdade de Engenharia, Universidade do Porto, 2006.

[Liou, 1999] Liou, S., Soelaeman, H., Leung, P., *Distance Learning Power Engineering Laboratory*. In: IEEE Computer Application in Power, 1999.

[Liu, 2006] Liu, B., Riemenschneider, S. e Xu, Y., *Gearbox fault diagnosis using empirical mode decomposition and Hilbert spectrm*. In Mechanical Systems and Signal Processing, Volume 20, Issue 3. Elsevier, 2006.

[**Lopes, 2007**] Lopes, S., *Laboratórios de acesso remoto em física*, dissertação de mestrado, Faculdade de Ciências e Tecnologia da Universidade de Coimbra, Coimbra, 2007.

[**Lucena, 2004**] Lucena C., Fuks H. *A Educação na Era da Internet*. Clube do Futuro, Rio de Janeiro Mathworks, 2004.

[**Mathworks**] Mathworks, *MATLAB*, <http://www.mathworks.com/products/Simulink/>.

[**Marchezan, 2004**] Marchezan, A. R.; Chella, M. T.; Ferreira, E. C. *Laboratório Remoto Aplicado ao Ensino de Engenharia Electrónica*. In: Workshop de Ciência da Computação e Sistemas da Informação da Região Sul – Workcompsul, Florianópolis, 2004.

[**NI**] National Instruments, <http://www.ni.com/>.

[**Nielsen, 1999**] Nielsen, J. *Design Web Usability*. New Riders Publish., Indianapolis, Indiana USA, 1999.

[**Odgaard, 2007**] P. F. Odgaard and M. V. Wickerhauser, *Karhunen-Loève (PCA) based detection of multiple oscillations in multiple measurement signals from large-scale process plants*, in Control Conference, 2007.

[**Ooi, 2008**] Su Ki Ooi and Erik Weyer, *Detection of oscillations in control loops in irrigation channels*, in Control Engineering Practice 16, 2008.

[**Pinto, 2009**] Pinto R., *Novas abordagens ao estudo de sinais boimédicos: Análise em tempo frequência e transformada de Hilbert-Huang*, dissertação de Doutorado, Universidade de Lisboa, 2009.

[**Sabin, 2007**] Sabin, W., *Discrete-Signal Analysis and Design*, A JOHN WILEY & SONS, INC, 2007.

[Samoila, 2007] Samoila, C., S. Cosh, D. Ursutiu, “Competences, Remote Labs and Bologna Process”, *In: Advances on remote laboratories and e-learning experiences*. L. Gomes and J. Garcia-Zubia (Ed), pp. 63-96. Univ. Deusto, Bilbao, 2007.

[Sánchez, 2004] Sánchez, J., Dormido, S., Pastor, R., Morilla, F., *A Java/MATLAB-Based Environment for Remote Control System Laboratories: Illustrated With an Inverted Pendulum*. In: IEEE Transactions on Education, 2004.

[Schafer, 2003] Schafer, T., Seigneur, J. M. e Donelly, A. *PEARL: A Generic Architecture for Live Experiments in a Remote Lab*, In Proc. Of International Conference on Simulation and Multimedia in Engineering Education, IEEE, 117-122, 2003.

[Schmid, 2000] Schmid, C., Ali, A.: *A web-based system for control engineering education*. In American Control Conference, Chicago, 2000.

[Shen, 1999] Shen H., Xu Z., Dalager B., Kristiansen V., Strøm Ø., Shur M. S., Fjeldly T. A., Lü J. e Ytterdal T. *Conducting Laboratory Experiments over the Internet*, IEEE transactions on education, vol. 42, no. 3, p.180-185, 1999.

[Souza, 2001] Souza, A., Oliveira, J., *Laboratórios acessíveis via Internet: um recurso didático para o ensino/aprendizado de engenharia eléctrica*. In: VII Encontro Ensino Em Engenharia, Brasil, 2001.

[Thornhill, 2000] N. F. Thornhill, S. L. Shah and B. Huang, *Detection and Diagnosis of Unit-Wide Oscillations*, in Process control and instrumentation, Glasgow, 2000.

[Viedma, 2005] Viedma, G., Dancy, I., Lundberg, K., *A ewb-based linear-systems iLab*. In: American Control Conference, Portland, 2005.

[Yeung, 2003] K. Yeung, J. Huan, *Development of a remote-access laboratory: a dc motor control experiment*, Computers in Industry, Vol. 52, pp. 305-311, 2003.

[Zeilmann, 2003] Zeilmann, R., *Web-based control experiments on a Foundation Fieldbus pilot plant*. In: IFAC International Conference on Fieldbus and Their Applications, Oxford, 2003.